ISSN 0252–9742

Bulletin

of the

European Association for Theoretical Computer Science

EATCS

Number 138 October 2022

Council of the European Association for Theoretical Computer Science

President:	Artur Czumaj	United Kingdom
Vice Presidents:	Anca Muscholl	France
	Giuseppe F. Italiano	Italy
Treasurer:	Jean-Francois Raskin	Belgium
Bulletin Editor:	Stefan Schmid	Germany

Ivona Bezakova	USA	Anca Muscholl	France
Tiziana Calamoneri	Italy	Luke Ong	UK
Thomas Colcombet	France	Tal Rabin	USA
Artur Czumaj	UK	Eva Rotenberg	Denmark
Javier Esparza	Germany	Maria Serna	Spain
Fabrizio Grandoni	Switzerland	Alexandra Silva	USA
Thore Husfeldt	Sweden, Denmark	Jiri Sgall	Czech Republic
Giuseppe F. Italiano	Italy	Ola Svensson	Switzerland
Fabian Kuhn	Germany	Jukka Suomela	Finland
Slawomir Lasota	Poland	Till Tantau	Germany
Elvira Mayordomo	Spain	Sophie Tison	France
Emanuela Merelli	Italy	Gerhard Wöeginger	The Netherlands

Past Presidents:

Maurice Nivat	(1972–1977)	Mike Paterson	(1977–1979)
Arto Salomaa	(1979–1985)	Grzegorz Rozenberg	(1985–1994)
Wilfred Brauer	(1994–1997)	Josep Díaz	(1997–2002)
Mogens Nielsen	(2002–2006)	Giorgio Ausiello	(2006–2009)
Burkhard Monien	(2009–2012)	Luca Aceto	(2012–2016)
Paul Spirakis	(2016–2020)		

Secretary Office:	Efi Chita	Greece
	Emanuela Merelli	Italy

EATCS Council Members

email addresses

Ivona Bezakova	ib@cs.rit.edu
Tiziana Calamoneri	calamo@di.uniroma1.it
Thomas Colcombet	Thomas.Colcombet@irif.fr
Artur Czumaj	A.Czumaj@warwick.ac.uk
Javier Esparza	esparza@in.tum.de
Fabrizio Grandoni	fabrizio@idsia.ch
Thore Husfeldt	thore@itu.dk
Giuseppe F. Italiano	giuseppe.italiano@uniroma2.it
Fabian Kuhn	kuhn@cs.uni-freiburg.de
Slawomir Lasota	sl@mimuw.edu.pl
Elvira Mayordomo	elvira@unizar.es
Emanuela Merelli	emanuela.merelli@unicam.it
Anca Muscholl	anca@labri.fr
Luke Ong	luke.Ong@cs.ox.a.uk
Tal Rabin	chair.sigact@sigact.acm.org
Jean-Francois Raskin	jraskin@ulb.ac.be
Eva Rotenberg	eva@rotenberg.dk
Maria Serna	mjserna@cs.upc.edu
Stefan Schmid	stefan.schmid@tu-berlin.de
Alexandra Silva	alexandra.silva@cornell.edu
Jiri Sgall	sgall@iuuk.mff.cuni.cz
Ola Svensson	Ola.Svensson@epfl.ch
Jukka Suomela	jukka.suomela@aalto.fi
Till Tantau	tantau@tcs.uni-luebeck.de
Sophie Tison	sophie.tison@lifl.fr
Gerhard Wöeginger	g.j.woeginger@math.utwente.nl

Bulletin Editor: Stefan Schmid, Berlin, Germany
Cartoons: DADARA, Amsterdam, The Netherlands

The bulletin is entirely typeset by pdfTEX and ConTEXt in TXfonts.

All contributions are to be sent electronically to

`bulletin@eatcs.org`

and must be prepared in LaTeX 2$_\varepsilon$ using the class `beatcs.cls` (a version of the standard LaTeX 2$_\varepsilon$ `article` class). All sources, including figures, and a reference PDF version must be bundled in a ZIP file.

Pictures are accepted in EPS, JPG, PNG, TIFF, MOV or, preferably, in PDF. Photographic reports from conferences must be arranged in ZIP files layed out according to the format described at the Bulletin's web site. Please, consult `http://www.eatcs.org/bulletin/howToSubmit.html`.

We regret we are unfortunately not able to accept submissions in other formats, or indeed submission not *strictly* adhering to the page and font layout set out in `beatcs.cls`. We shall also not be able to include contributions not typeset at camera-ready quality.

The details can be found at `http://www.eatcs.org/bulletin`, including class files, their documentation, and guidelines to deal with things such as pictures and overfull boxes. When in doubt, email `bulletin@eatcs.org`.

Deadlines for submissions of reports are January, May and September 15th, respectively for the February, June and October issues. Editorial decisions about submitted technical contributions will normally be made in 6/8 weeks. Accepted papers will appear in print as soon as possible thereafter.

The Editor welcomes proposals for surveys, tutorials, and thematic issues of the Bulletin dedicated to currently hot topics, as well as suggestions for new regular sections.

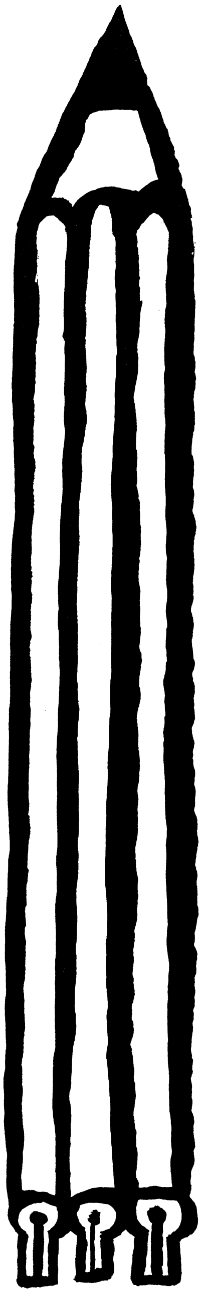
The EATCS home page is `http://www.eatcs.org`

Table of Contents

EATCS MATTERS

EATCS COLUMNS

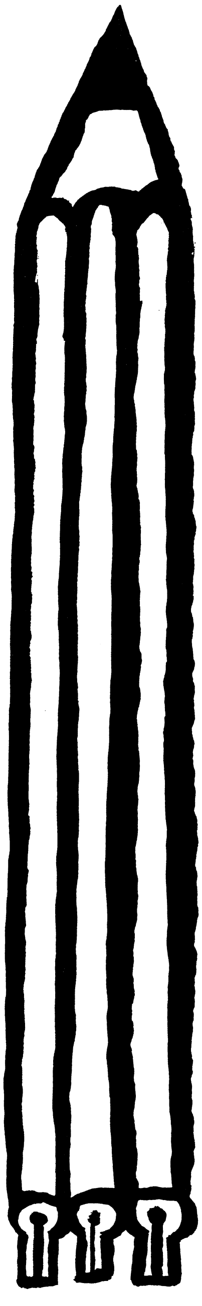
EATCS Matters



Dear EATCS members,

I hope my letter finds you and your family safe and in good health. While some of us may still have impact of the global coronavirus pandemic, I hope the situation is improving and is coming back to normal, or at least similar to what we have seen a few years back. I also hope that the last two years will allow to think how to effectively enhance our online and hybrid research experience. Already this spring and later in the summer we have seen many research activities taking place in person or in the hybrid mode, and a lot of active scientific interaction. And most importantly, we see some fantastic research done by our community; and so I take the opportunity to wish you all the best and much success for your work.

EATCS ICALP 2022, the EATCS flagship conference, was run this July in the hybrid format, after two years of having ICALP run online only. As always, the conference had an impressive scientific program highlighting the strength of the research across many areas within theoretical computer science. On behalf of the entire community and the EATCS I would like to thank the Programme Committee led by the chairs David Woodruff and Mikołaj Bojańczyk, the organizers Paris, led by Thomas Colcombet (who have been making special efforts to make the conference environment friendly), and especially - all the participants, for their fantastic efforts that helped to make the ICALP 2022 conference a great success. Furthermore, ICALP 2022 was the occasion to celebrate



the 50th anniversary of both EATCS and the first ICALP, which was first held in 1972 in Rocquencourt, in the Paris area. We plan to have a detailed report of ICALP 2022 in the next issue of the Bulletin.

We also had very successful three EATCS partner conferences: ESA 2022, MFCS 2022, and DISC 2022. While some of the activities in these conferences have been online, all these conferences have been well attended in-person.

In this issue of the Bulletin, you will find the calls for nominations for the EATCS Award, the Presburger Award, the EATCS Distinguished Dissertation Award, and the EATCS Fellows. As usual, we are lucky to have very strong committees for each of the awards, and I thank all the award committee members in advance for their important service. I strongly encourage you to send nominations for these prestigious awards. I am aware of the fact that we are all very busy and that it takes time and efforts to prepare strong nominations, but our best researchers and best papers can only win awards if they are nominated. Moreover, awards put areas of, as well as inspirational figures in, theoretical computer science in the spotlight and can serve to inspire young researchers. I look forward to seeing who the award winners will be and to working with all of you to make the EATCS even more influential than it already is.

As usual, the October issue of the Bulletin has the first Call for Papers for ICALP 2023, the flagship conference of the EATCS and an important meeting of the theoretical computer science community world-wide. The 50th EATCS International Colloquium on

Automata, Languages, and Programming (ICALP 2023), will be held on July 10-14, 2023 in Paderborn, Germany. After two years of online events, and in this 2022 year being in the hybrid format, we hope that ICALP 2023 will be mostly an in-person conference, with the talks planned to be delivered in person; we hope for a very good in-place audience. We have a great list of invited speaker and expect a fantastic scientific program selected by the PCs led by the chairs Uriel Feige and Kousha Etessami. Furthermore, this will be the 50th edition of ICALP and we hope this to be a great occasion to celebrate ICALP and its impact on the Theory community. I will write more details about the planned activities in the next issue of the Bulletin, but as for now: please pencil these dates in your diary and I hope to see many of you joining us in these celebrations.

As usual, let me close this letter by reminding you that you are always most welcome to send me your comments, criticisms and suggestions for improving the impact of the EATCS on the Theoretical Computer Science community at president@eatcs.org. We will consider all your suggestions and criticisms carefully.

I look forward to seeing many of you around, in-person or online, and to discussing ways of improving the impact of the EATCS within the theoretical computer science community.

Artur Czumaj
University of Warwick, UK
President of EATCS

president@eatcs.org

October 2023

Dear EATCS community,

the last issue of this year 2022 contains several interesting articles and contributions, including an interview with Monika Henzinger (see the "Know the person behind the papers" column) and a viewpoint column by Sophie Huiberts who examines how awards' recipients are selected in our communities and how this relates to the status of women in our field. Omer Reingold, in the theory blogs column, shares his thoughts on how to make a research environment supportive for young researchers, and talks about his "research life stories" project and experiences with starting several blogs. In the logic column, an elegant proof is presented, accessible to the non-experts, about a fundamental result related to the question whether there is a logic that captures PTIME. We further have two interesting surveys in the algorithms and the complexity column (related to Taylor's theorem, and about derandomizing space-bounded computation, respectively), an overview of current challenges and open questions in the area of robust clock synchronization (in the distributed computing column), and an assessment of interactive online learning systems (in the educational column). Finally, this issue contains several conference reports and related statistics.

I would like to thank all the authors and contributors of this issue. On this occasion, also a reminder that if you have an interesting new direction to share, or a summary of important open problems in areas

of interest, please do not hesitate to contact the corresponding area editor or me any time, we are always open to contributions to the Bulletin.

Enjoy the new Bulletin and I wish everyone a nice remainder of the year!

Stefan Schmid, Berlin
October 2022

The EATCS Award 2023

Call for Nominations

Deadline: January 15, 2023

The European Association for Theoretical Computer Science (EATCS) annually honours a respected scientist from our community with the prestigious EATCS Distinguished Achievement Award. The award is given to acknowledge extensive and widely recognized contributions to theoretical computer science over a life long scientific career. For the EATCS Award 2023, candidates may be nominated to the Awards Committee chaired by Johan Håstad.

Nominations will be kept strictly confidential. They should include supporting justification and be sent by e-mail to the chair of the EATCS Award Committee:

Johan Håstad
`eatcs-award@eatcs.org`

by January 15, 2023. Previous recipients of the EATCS Award are:

R.M. Karp (2000)
C. Böhm (2001)
M. Nivat (2002)
G. Rozenberg (2003)
A. Salomaa (2004)
R. Milner (2005)
M. Paterson (2006)
D.S. Scott (2007)
L.G. Valiant (2008)
G. Huet (2009)
K. Mehlhorn (2010)
B. Trakhtenbrot (2011)
M.Y. Vardi (2012)
M.E. Dyer (2013)
G.D. Plotkin (2014)
C. Papadimitriou (2015)
D. Kozen(2016)
É. Tardos(2017)
N. Nisan (2018)
T. Henzinger (2019)
Mihalis Yannakakis(2020)
T. Pitassi (2021)
P. Cousot (2022)

The Award will be assigned during a ceremony that will take place during ICALP 2023 in Paderborn (https://icalp2023.cs.upb.de/).

The Presburger Award for Young Scientists 2023

Call for Nominations

Deadline: 15 February 2023

The Presburger Award recognises outstanding contributions by a young scientist in theoretical computer science, documented by a published paper or a series of published papers. It is named after Mojzesz Presburger who accomplished his ground-breaking work on decidability of the theory of addition (known today as Presburger arithmetic) as a student in 1929. The award is conferred annually by the European Association for Theoretical Computer Science (EATCS) at the International Colloquium on Automata, Languages, and Programming (ICALP).

Nominated scientists can be at most 35 years old on January 1st of the year of the award. Thus, for the 2023 award, the nominee should be born in 1987 or later. Nominations for the Presburger Award can be submitted by any member or group of members of the theoretical computer science community, but not by the nominee themselves nor the advisors for their master's thesis or doctoral dissertation.

The Presburger Award committee for 2023 consists of Mikołaj Bojanczyk (University of Warsaw, chair), Uriel Feige (The Weizmann Institute), and Tal Malkin (Columbia University). Nominations, consisting of a two page justification and (links to) the relevant publications, as well as additional supporting letters, should be sent by e-mail to:

`presburger-award@eatcs.org`

The subject line of every nomination should start with *Presburger Award 2023*, and the message must be received before **February 15th, 2023**.

The award includes an amount of 1000 Euro and an invitation to ICALP 2023 for a lecture.

Previous Winners:

Mikołaj Bojańczyk, 2010
Patricia Bouyer-Decitre, 2011
Venkatesan Guruswami, 2012
Mihai Pătraşcu, 2012
Erik Demaine, 2013
David Woodruff, 2014
Xi Chen, 2015
Mark Braverman, 2016
Alexandra Silva, 2017
Aleksander Mądry, 2018
Karl Bringmann, 2019
Kasper Green Larsen, 2019
Dmitriy Zhuk, 2020
Shayan Oveis Gharan, 2021
Dor Minzer, 2022

Official website: `http://www.eatcs.org/index.php/presburger`

EATCS Distinguished Dissertation Award 2022

Call for Nominations

Deadline: January 10, 2023

The EATCS establishes the Distinguished Dissertation Award to promote and recognize outstanding dissertations in the field of Theoretical Computer Science. Any PhD dissertation in the field of Theoretical Computer Science that has been successfully defended in 2022 is eligible. Up to three dissertations will be selected by the committee for year 2022. The dissertations will be evaluated on the basis of originality and potential impact on their respective fields and on Theoretical Computer Science. Each of the selected dissertations will receive a prize of 1000 Euro. The award receiving dissertations will be published on the EATCS web site, where all the EATCS Distinguished Dissertations will be collected.

The dissertation must be submitted by the author as an attachment to an email message sent to the address dissertation-award@eatcs.org with subject EATCS Distinguished Dissertation Award 2022 by January 10, 2023. The body of the message must specify: Name and email address of the candidate; Title of the dissertation; Department that has awarded the PhD and denomination of the PhD program; Name and email address of the thesis supervisor; Date of the successful defense of the thesis.

A five page abstract of the dissertation and a letter by the thesis supervisor certifying that the thesis has been successfully defended must also be included. In addition, an endorsement letter from the thesis supervisor, and possibly one more endorsement letter, must be sent by the endorsers as attachments to an email message sent to the address dissertation-award@eatcs.org with subject EATCS DDA 2022 endorsement. The name of the candidate should be clearly specified in the message.

The dissertations will be selected by the following committee:

- Susanne Albers (chair)
- Elvira Mayordomo
- Jaroslav Nešetřil
- Damian Niwiński
- Vladimiro Sassone
- Alexandra Silva
- David Woodruff

The award committee will solicit the opinion of members of the research community as appropriate. Theses supervised by members of the selection committee are not eligible. The EATCS is committed to equal opportunities, and welcomes submissions of outstanding theses from all authors.

EATCS-Fellows 2023

Call for Nominations

Deadline: January 31, 2023

The EATCS Fellows Program is established by the Association to recognize outstanding EATCS Members for their scientific achievements in the field of Theoretical Computer Science. The Fellow status is conferred by the EATCS Fellows-Selection Committee upon a person having a track record of intellectual and organizational leadership within the EATCS community. Fellows are expected to be "model citizens" of the TCS community, helping to develop the standing of TCS beyond the frontiers of the community.
In order to be considered by the EATCS Fellows-Selection Committee, candidates must be nominated by at least four EATCS Members. Please verify your membership at www.eatcs.org.

The EATCS Fellows-Selection Committee consists of

- Christel Baier
- Mikołaj Bojańczyk
- Mariangiola Dezani
- Josep Diaz
- Giuseppe F. Italiano

INSTRUCTIONS:

Proposals for Fellow consideration in 2023 should be submitted by January 31, 2023 by email to the EATCS Secretary (secretary@eatcs.org). The subject line of the email should read
"EATCS Fellow Nomination - <surname of candidate>"
A nomination should consist of details on the items below. It can be co-signed

by several EATCS members. Two nomination letters per candidate are recommended. If you are supporting the nomination from within the candidate's field of expertise, it is expected that you will be specific about the individual's technical contributions.

To be considered, nominations for 2023 must be received by January 31, 2023.
1. Name of candidate Candidate's current affiliation and position Candidate's email address, postal address and phone number Nominator(s) relationship to the candidate
2. Short summary of candidate's accomplishments (citation – 25 words or less)
3. Candidate's accomplishments: Identify the most important contributions that qualify the candidate for the rank of EATCS Fellow according to the following two categories:
A) Technical achievements
B) Outstanding service to the TCS community Please limit your comments to at most three pages.
4. Nominator(s):
Name(s) Affiliation(s), email and postal address(es), phone number(s)
Please note: all nominees and nominators must be EATCS Members.

Institutional Sponsors

CTI, Computer Technology Institute & Press "Diophantus"
Patras, Greece

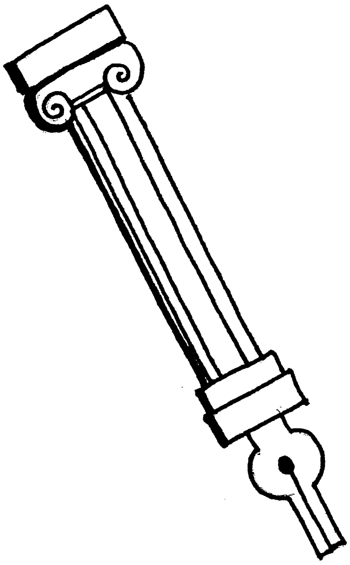
CWI, Centum Wiskunde & Informatica
Amsterdam, The Netherlands

MADALGO, Center for Massive Data Algorithmics
Aarhus, Denmark

Microsoft Research Cambridge
Cambridge, United Kingdom

Springer-Verlag
Heidelberg, Germany

EATCS
Columns



The Interview Column

by

Chen Avin and Stefan Schmid

Ben Gurion University, Israel and TU Berlin, Germany
{chenavin,schmiste}@gmail.com

Know the Person behind the Papers

Today: Monika Henzinger

Bio: *Monika Henzinger is a Professor at the University of Vienna, Austria, heading the research group of Theory and Applications of Algorithms. She received her PhD in 1993 from Princeton University and was an assistant professor at Cornell University, a researcher at Digital Equipment Corporation, the Director of Research at Google and a professor at EPFL, Switzerland, before moving to the University of Vienna. Professor Henzinger received a Dr. h. c. degree from the Technical University of Dortmund, Germany, two ERC Advanced Grants (2014 and 2021), the Wittgensteinpreis of the Austrian Science Foundation, the Carus Medal of the German Academy of Sciences, a SIGIR Test of Time Award, a Netidee SCIENCE Award of the Internet Foundation Austria, a European Young Investigator Award, an NSF CAREER Award, and a Top 25 Women on the Web Award. She is a fellow of the ACM and of the EATCS and a member of the Austrian Academy of Sciences and the German Academy of Sciences Leopoldina. She is an editor of the Journal of the ACM and the SIAM Journal on Computing and a member of the Swiss Science Council.*

EATCS: We ask all interviewees to share a photo with us. Can you please tell us a little bit more about the photo you shared?

MH: I like this picture from Wikipedia - I was on a panel of the Austrian Science Foundation and didn't even notice when this picture was taken. In the panel I was describing how much the German Academic Scholarship Foundation has helped me to decide that I want to become a researcher and in the meantime the Austrian Academy of Sciences has started an Austrian Academic Scholarship Foundation. I think such a foundation is an excellent tool to inspire students to pursue an academic career.

EATCS: Can you please tell us something about you that probably most of the readers of your papers don't know?

MH: Starting spring 2023 I will transfer as professor to IST Austria.

EATCS: Is there a paper which influenced you particularly, and which you recommend other community members to read?

MH: As a Phd student I particularly liked the paper on Amortized Computational Complexity by my PhD advisor, Bob Tarjan.

EATCS: Is there a paper of your own you like to recommend the readers to study? What is the story behind this paper?

MH: This is the hardest question for me to answer. The paper with the best story is certainly the paper Faster Shortest-Path Algorithms for Planar Graphs. J. Comput. Syst. Sci. 55(1): 3-23 (1997) with Philip N. Klein, Satish Rao, and Sairam Subramanian. We proved the main result on the phone (it was 1996!) working through the night, the night before the STOC deadline. We showed how to exploit the power of the planar separator theorem to give a linear-time algorithm for shortest paths in graphs. It sparked a lot of follow-up work.

EATCS: When (or where) is your most productive working time (or place)?

MH: I work best in the morning and need a quiet place to work. That's why I love to work from home.

EATCS: What do you do when you get stuck with a research problem? How do you deal with failures?

MH: I revisit problems that I get stuck on periodically, sometimes with a different PhD student or collaborator than before. But I don't consider that a failure - if it didn't happen, I would worry that the problems I work on are not ambitious enough. Instead, I consider it a failure to write a paper that is not read. (As I don't know what other researchers read, I use the number of citations as a potentially poor replacement.)

EATCS: Is there a nice anecdote from your career you like to share with our readers?

MH: In high school I liked all sciences and I really didn't know what to study. It was my mother, who was not an academic, but who saw my passion for programming, she was the one who advised me to study Computer Science. Luckily, I listened to her.

EATCS: Do you have any advice for young researchers? In what should they invest time, what should they avoid?

MH: Work on problems that you think will be important for the future.

EATCS: What are the most important features you look for when searching for graduate students?

MH: I look for students who are highly motivated to work on algorithms and are creative thinkers.

EATCS: Do you see a main challenge or opportunity for theoretical computer scientists for the near future?

MH: Staying relevant. I am concerned that a large part of our community is working too much on problems that the community likes and too little on problems that have an impact in other areas of Computer Science or other fields. However, on the positive side, there is also a part of our community that is really concerned about impact.

EATCS: How was your research affected by the pandemic? How do you think it will affect us as a community?

MH: As I said above I like working from home and, thus, the pandemic was helpful for my research. It also increased the online research collaboration, even between continents. I just hope that conferences will stay hybrid as it enables researchers who, for various reasons, cannot travel (for example because they have small children) to participate in conferences at least online.

Please complete the following sentences?

- *My favorite movie is...* no specific one, but I like movies that make me laugh.
- *Being a researcher...* is a vocation for me, one of my favorite things to do.
- *My first research discovery...* was a lower bound for a problem related to the dictionary problem. It was my master thesis which was supervised by Kurt Mehlhorn and the collaboration with him got me hooked on research.

The Viewpoint Column

by

Stefan Schmid
TU Berlin, Germany
`stefan.schmid@tu-berlin.de`

Prizes and Prejudice

Sophie Huiberts
Department of Computer Science, Columbia University, USA

Abstract

In academia we have a wide assortment of awards, but we have even more work deserving of such recognition. What work and which people get recognized has significant influence on who gets hired and who gets funded. I examine how awards' recipients are selected in our communities and how this relates to the status of women in our field.

When we as academics give a prize or award to a person or group, this serves to uplift the recipient and to indicate that their work is valued in the community. This signal of appreciation then benefits the recipient and their area of research; a best poster award can be a nice boost to the CV of an early career researcher, while the Gödel Prize commands the admiration of colleagues, future PhD candidates, and higher-ups in our own universities. Although awards are meant to reflect existing appreciation for certain projects and people, they also create and reinforce this appreciation in turn.

Every year there is a lot of work deserving of recognition, and only so many awards to go around. This implies that scientific merit alone is not enough to deterimine a winner, hence other considerations will inevitably play a part. What are these other factors of influcence and what is their impact on our discipline? While there are many valid answers to this question worth examining, we will focus on gender in this article. The phenomena discussed here are not unique to gender; we could find and replace the gendered qualifiers in this text by a variety of others and a very similar discussion can be held.

Most awards in our discipline are superficially gender-neutral: no formal rule forbids them from being given to researchers of certain genders. Despite this formal pretense, I argue that, in practice, many awards serve to maintain the gendered status quo in which women are at a serious disadvantage compared to men. To illustrate this, I will describe three prizes, big and small, and how I perceive them as unfairly elevating already-privileged researchers over their minoritized peers.

I want to emphasize that this article is intended as a complaint towards those with the power to affect prize-giving. There is no doubt in my mind that the recipients of the prizes mentioned have each done high-quality work that is worthy of the recognition they received.

1 “This prize is too small to worry about inclusion”

In my home country of the Netherlands, we have an organization uniting discrete math, algorithms, algebra and number theory, and every year they organize two symposia. At these events, there are always many PhD candidates speaking. At the end of the symposium, a ‘best PhD student presentation prize’ is handed out by a famous Dutch mathematician and a photo of the prize ceremony is published in the Dutch mathematical trade magazine.

The prize committee consists of just this one mathematician; let’s call him FM for his role as Famous Mathematician. If you talk about FM to anyone who is not an old man, you will soon learn of his documented history of egregious sexist remarks. Thus, it should not be a surprise to learn that all recipients in the prize’s history have been men, in sharp contrast to the population of PhD speakers.[1]

At this point you might be asking ‘why is this allowed to happen? Why does the symposium give FM its platform and the time and attention of its attendees? Don’t the organizers know about this guy’s reputation?’ If so, then you are in luck, because I asked these exact questions to one of the organizers.

The organizer I spoke to is an old man. Throughout our conversation, I got the impression that he did not know about FM’s reputation and he seemed unwilling to believe the stories about FM’s sexism when I mentioned those. (I later learned that I was not the first person to mention this to him.) And anyway, I was told, this is not a serious prize, nobody pays attention to it, so no big deal even if it were biased.

This last point seems especially callous, considering the impact of prizes on an early-career researcher’s CV. Every recipient whose CV I found online mentioned receiving the “Famous Mathematician PhD Presentation Prize”. Without directing blame at the awardees, we can observe that this prize serves to launder FM’s sexism and turn it into a respectable line on the CVs of selected men.

What this story illustrates is that there are bad actors in our communities, that the people in power fail to recognize them as such, and that this system results in a concrete career advantage for men over anyone else.

2 “This prize has gender-oblivious rules”

The Turing Award is embarrassing, and all CS researchers ought to feel ashamed about our professional association, the ACM, allowing this award to exist in its current form. The last woman to win a Turing Award was Shafi Goldwasser, a decade ago, after whom fifteen men have gone on to win this prize. In its 56 year

[1] If your immediate reaction was to think that perhaps the men were all better speakers, I encourage you to reflect on that impulse and consider if this might be your biases talking.

history, we can count three women and one hundred and twenty-eight men among the recipients.

This problem extends far beyond the committee making the final call: the bias is built into every step of the process. Candidates for the Turing Award are generated from nominations from the computing community, and every year many nominations are received by the ACM. However, the committee receives only about one woman nominee every five years.[2] That is not a typo: one woman nominee every five years. The theory community alone could easily nominate dozens of excellent women every year, each a deserving Turing Award recipient. Let's not mince words here: anyone who believes the Turing Award to be worth anything, who was around to nominate people over the past decades, and who failed to nominate a woman, is at fault.

Neither are the ACM and its award committee free from blame. It is beyond obvious that the current nomination system is not functioning and must be completely reworked. Significant action is required from ACM leadership in order to make the Turing Award capable of being anywhere near equitable. At this moment the Turing Award is failing all of us and harming our discipline. For as long as the nomination and selection procedures are not overhauled, we must recognize the Turing Award for what it is: not a prize for excellent researchers in general, but primarily a prize for excellent *male* researchers.

3 "This prize is for senior researchers"

Today, there are disproportionately many men in theoretical computer science. This is bad and frustrating to those of marginalized genders, both those in the field right now and those who might enter the field in the future. However, the ratio is even more skewed among senior researchers, reflecting the fact that the situation was even worse in decades past.

This ought to prompt caution in those who institute prizes for senior researchers. One example of such a prize is a Test of Time (ToT) Award: awarded to the best papers published in a venue 10, 20 or 30 years ago. FOCS introduced its ToT Award in 2019, whereas STOC introduced its ToT Award in 2021. These recent innovations can be expected to, at best, reflect the gender bias of the past and reproduce it in the distribution of power in the present day. At worst, we can expect outcomes like the ToT Awards of FOCS 2019, where I count awards for ten men and zero women. In a time when more women are entering the field than ever before, I question whether it is appropriate for the community to institute awards that can only be given to senior researchers. What benefit is created by such an

[2] Source: `https://youtu.be/lJtdOsjy59A?t=5560`

award, and what harm might it cause?

That said, I observe that the ToT Awards have been getting better over time. As mentioned, I counted ten men and zero women recipients for FOCS 2019, but for FOCS 2021 this goes up to fifteen men and two women. While the playing field is still far from equal, I believe that this shift signals an awareness of the problem and a willingness to improve our situation.

4 Closing remarks

Today we learned that prizes not just reflect attitudes from the past and present, but also have part in shaping those of the future. The consequences of this were illustrated by way of three prizes — three out of many — with a history of ignoring women's contributions to the field. Our field is in a bad place when it comes to diversity, and our prizes are not setting us up for a brighter future.

Removing bad actors from their power is a necessary step towards stopping this ongoing harm to our communities, but it is merely the most obvious measure we can take. We are all part of this system producing bad outcomes, and we can not afford to believe the fairy tale that superficial gender-neutrality is the way out.

I finish with a call to action. For everyone, I suggest to make a list of people who would deserve a prize nomination or two. Leave out any men, and keep going until you have at least a few dozen names. The next time you are in a position to nominate anyone for a prize, award or fellowship, use the list. For anyone on a prize committee, I urge you to study which factors contribute to unfair biases and work with your fellow committee members to counteract these forces. There is plenty of literature out there which can guide you on this path. And anyone who is hiring or otherwise in a position to judge people on the basis of their CVs, I ask you to recognize the reality that the presence or absence of signals of prestige reflects much more than academic merit alone. Let this recognition inform your decisions going forward.

The Theory Blogs Column

by

Luca Trevisan

Bocconi University
Via Sarfatti 25, 20136 Milano, Italy
L.Trevisan@UniBocconi.it
`https://lucatrevisan.github.io`

In this issue, Omer Reingold talks about his experience starting not just one, not even two, but three group blogs on theoretical computer science.

Omer is well-known for his price-winning research on the foundations of cryptography, on computational complexity, on combinatorics, and on fairness in artificial intelligence, but he is also an extremely caring and thoughtful mentor. He has been thinking for a long time about what makes a research environment welcoming and supportive for young researchers, and about how to foster such an environment.

In his guest column, Omer tells us about his experiences with theory blogs, including his “research life stories" project.

BRINGING RESEARCH-LIFE TO CENTERSTAGE

Omer Reingold

Like many in our community, I learn a lot from theory blogs. But earlier in my career I couldn't imagine that I will become a blogger myself. Planning aside, by now, I have founded and managed three research blogs – "Windows on Theory" in Microsoft Research Silicon Valley, "Theory Dish" in Stanford's theory group and recently "TOC 4 Fairness" as part of Simons Foundation's collaboration on Algorithmic Fairness. So, what changed? The first reason for which I thought I couldn't blog is that I didn't perceive myself as being enough of an exhibitionist to be a blogger. Turns out that this is much less of a problem than I'd like to think. In fact, with age, I am even more excited to talk about, hmmm, myself (so thank you Luca for this excellent opportunity). In addition, I always saw the value of blogs for the communication of ideas within a discipline and as a powerful tool for popularizing science. But I also always felt that I am too busy, that I write too slowly and that I will have enough time to focus on popular writing once I get tired of research. What convinced me to take the plunge is the wonderful theory group that existed in Microsoft Research Silicon Valley and whose brilliance I wanted to share with the theory community at large. It's not a coincidence that my final blog post on "Windows on Theory," before living the blog in Boaz Barak's most capable hands[1] was titled "A Social Blogger." Blogging for me is something to do with a community and for a community.

Science and Scientists

The group blogs that I formed contain a mix of scientific and meta-scientific posts. At any given point, I (as many others, I am sure) have several scientific insights that I'd like to share with the community more directly than in research papers. At times I enjoyed posting about these insights[2] but, unfortunately, to many of those I will never get (a thinly stretched professor and a slow writer, remember?) I do have a better track record in convincing others to blog about science. In particular, the theory group at Stanford now allows blog-writing to be a possible outcome of our quals (in addition to an oral presentation). We believe that this could be an excellent capability for our students to develop and a good scientific service.

The posts I find more time to write are meta-scientific. How should our conferences operate?[3] How to run a successful program committee?[4] How obsessing on the shortcomings

[1] I always considered the recruiting of Boaz to be my most important contribution to TOC blogs.

[2] A few of my favorite posts on that front are: "Occupy Database – Privacy is a Social Choice," "Rigged Lottery, Bible Codes, and Spinning Globes: What Would Kolmogorov Say?" "Advanced Studies in Estate Management: He Who Was Married to Three Women," and discussions of other's research in celebration of their awards as in "2012 Turing to Goldwasser and Micali."

[3] In various Windows on Theory and Theory Dish posts including "FOCS/STOC: Protect the Venue, Reform the Meeting,"'with Boaz, suggesting a reform of our flagship conferences, "Can We Get Serious?," which criticizes the chosen path as well as others on page limits and anonymous submissions.

[4] For example, in my Windows on Theory posts "Some Reflections on the FOCS PC Work"

of the community may obscure its incredible successes.[5] The relationship of research with industry and society at large.[6] Scientific communication in relation to literature.[7] And also various exciting announcements.[8] At times, I felt like my voice had some positive impact on the community. At times I felt like the dog that barks while the caravan moves on. At times, I shared the stage with others with whom I disagreed. One way or another, I believe that throughout my career, I got (and am still getting) more than a fair share of influence on the TOC community that I love so much. I feel good about letting others take the lead.

If the previous two categories of posts are ones that I will likely continue contributing to on occasion, there is a third category that I am really passionate about. This category is not about the content of our science and not focused on the management and politics of science (but is often connected). The discussion I would like to promote in our blogs, in our conferences, in our universities and every other place where we "exist" is about the human aspect of doing research. Every social and emotional issue we often expect our community members to deal with "in their own time" or in their own personal support systems. I wholeheartedly believe that all of these should be explicit in our discussions, as science cannot be separated from the scientists, who are, at least for now, human. I also believe that the training of scientists should cover relevant social and psychological topics that could assist us in our own careers and when we mentor others. Uri Alon, a Weizmann professor of Biology, a friend, and one of my sources of inspiration in this quest, contrasted the significant amount of training one gets to optimize the usage of a fancy piece of equipment purchased for the lab with the absence of training to optimize the conditions for success of a student or a postdoc one mentors.

The Research-Life Stories Project

My first post ever, titled "Labor of Love," was a sign of what's to come. It talked about the different motivations that may lead to a research career and how they can change over time. But the project that expressed my conviction more than all was the research-life stories project in Windows on Theory (and to some extent also a career-advice project in Theory Dish). The call for stories was simple:

> "Please share with us events you remember from your research life."

The focus on stories was influenced by my experiences with a form of theater known as playback theater (which seems to be quite popular amongst theoretical computer scientists). The simple phrasing of the question was influenced by some studies in the field of education and was meant to not impose my preconceptions of what are the big issues that face people in their research careers. I encourage the readers to pause and think for themselves which events they would share if asked this question by a friend. And I'd like to emphatically assert that whether you are making your first steps in research or you are already retired, you have meaningful and important stories to share.

[5] In my Theory Dish post titled "TOC: a Personal Perspective (2021)."

[6] In my Theory Dish posts titled "The Research that Would Frustrate the Facebooks," "Pride and Prejudice: From Research to Practice" and "The 'Technologists' and Society."

[7] In my Theory Dish post titled "What's Your Story?"

[8] Like my Theory Dish post titled "TOC for Society" announcing the creation of FORC.

The project was very rewarding and I got many stories from people I admire about many chapters of their career (and wrote some myself). But I know that there are numerous additional stories to learn from and I am committed to uncover some more of them. For example, one segment that I didn't get enough stories from were people in the later stages of their careers. I believe that these stories could be extremely valuable for people in middle stages who experience a set of challenges that are not often talked about.

Shared and Unique Experiences

One of the major hope for the research-life stories project (fulfilled to a significant extent) was that it will expose to researchers in the beginning of their careers (for example, students) that the challenges that they are struggling with have been shared by many others in the community, including people that became very successful. Realizing that you are not alone with your experiences and that they don't say anything negative about you could be a very powerful experience. It can relieve some of the fundamental loneliness we sometimes experience. Quoting from the same post I then said:

> "A research career is different from most other jobs in its characteristic and challenges: Long period of education and training which is packed with uncertainty (Am I good enough? Will all this effort be rewarded by a suitable position in a suitable location to live in?), the tension between collaboration and competition, preserving creativity and relevance along the decades. To all of these and more, we should add that our community is so dispersed. Our collaborators, our audience, our points of reference, are not only the colleagues next door but probably more so our colleagues across the globe."

Of course, we are also all unique and our experiences are unique. Our upbringing, the different parts of our identity, our family conditions, our medical and psychological conditions and more are all affecting the reality and perception of our research life (as well as every other aspect of our life). Still, with all of our uniqueness, it can be comforting to know that in some ways we are also the same.

Of special importance is acknowledging that some groups of individuals within our community have another major layer in their research-life experience. Since our ability to understand the other is limited by our own experience, it is important to give room and directly listen to members of under-represented groups. In this respect, I want to acknowledge that one of the inspirations for the research-life stories project was Luca Trevisan's Turing Centennial posts in his blog "In Theory." In a tribute to Turing's life, Luca invited a sequence of inspiring posts from LGBTQ colleagues.

As for me, I cannot say that I have done enough but I never regretted anything I did to highlight or facilitate the voices that are not always well represented. Possibly, my most consistent effort (far from sufficient but still) was with respect to sexual misconduct. Among the relevant posts, perhaps the one I cherish the most is the Windows on Theory post titled "On intellectual passion and its unfortunate confusion with sexual passion (and how it may relate to issues of gender)." This is a translation from Hebrew by Oded Goldreich of (parts of) a post by an anonymous female graduate student in Humanities. I found it illuminating, especially in comparison with the discourse back in 2013, which was often less subtle. It demonstrate how basic freedoms and experiences that some of us get to take for granted are often denied from some of our colleagues. If you want to read one of the posts I mentioned here, perhaps this should be it.

What's next?

I have spent enough time in California, and listened enough times to Yusuf Islam (when he was still known as Cat Stevens) singing "but I might die tonight," to know that there is no "next" just "now." Well, change can start *now.* We can be aware of the "life" in our own research-life and attend to it now. We can foster discussions of the human aspects of research in meetings, in blogs, in courses [9], and in small steps that can happen now in the small corners of our research world or on the bigger stages that are sometimes offered to us.

[9] My Stanford course "The Practice of Theory Research" is an attempt in this direction.

The Algorithmics Column

by

Thomas Erlebach

Department of Computer Science
Durham University
Upper Mountjoy Campus, Stockton Road, Durham, DH1 3LE, UK
thomas.erlebach@durham.ac.uk

Approximate counting using Taylor's theorem: a survey

Viresh Patel[*] Guus Regts[†]

Abstract

In this article we consider certain well-known polynomials associated with graphs including the independence polynomial and the chromatic polynomial. These polynomials count certain objects in graphs: independent sets in the case of the independence polynomial and proper colourings in the case of the chromatic polynomial. They also have interpretations as partition functions in statistical physics.

The algorithmic problem of (approximately) computing these types of polynomials has been studied for close to 50 years, especially using Markov chain techniques. Around eight years ago, Barvinok devised a new algorithmic approach based on Taylor's theorem for computing the permanent of certain matrices, and the approach has been applied to various graph polynomials since then. This article is intended as a gentle introduction to the approach as well as a partial survey of associated techniques and results.

Keywords: approximate counting, independence polynomial, complex zeros, chromatic polynomial.

1 Introduction

Computational counting is an area of theoretical computer science, which, at its heart, is concerned with the computational problem of counting certain structures inside some combinatorial object given as input. Think of counting the number of satisfying assignments of some logical formula, or the number of independent sets in a graph. Often the structures to be counted have some natural weighting and one is interested in the weighted count.

In this article, we focus on graph counting problems and in particular on finding efficient algorithms for (approximately) counting objects of interest inside some input graph. The counting problems we consider here are ones where the number of objects

[*] School of Mathematical Sciences, Queen Mary, University of London. Email: viresh.patel@qmul.ac.uk.

[†] Korteweg de Vries Institute for Mathematics, University of Amsterdam. Email: guusregts@gmail.com. Funded by the Netherlands Organisation of Scientific Research (NWO): VI.Vidi.193.068

to be counted is typically super-polynomial in the size of the graph and cannot be directly enumerated in polynomial time. For example, the number of independent sets of a graph can be exponentially large in the size of the graph[1] and indeed, the problem of (approximately) counting independent sets of a graph is a rich area of research (here an independent set in a graph is a subset of vertices no two of which are adjacent). The problem of exact counting is often computationally hard (this is the case for independent sets [78, 86, 42]) so one is usually interested in approximation algorithms for counting problems. A notable exception is the problem of counting spanning trees of a graph. The number of spanning trees is typically exponential in the size of the graph, but spanning trees can be counted in polynomial time via the matrix tree theorem [62]. Throughout the article we use the example of counting independent sets in graphs to illustrate the various ideas we discuss. In fact the ideas apply more generally for counting many other graph theoretic objects including trees, matchings, cuts, and proper colourings (we discuss proper colourings towards the end of the article).

The basic combinatorial counting problems are often not treated directly, but are considered in more generality by examining their corresponding generating functions. For example, for independent sets, one is interested in the independence polynomial, which for a graph $G = (V, E)$, is defined to be the polynomial

$$Z_G(\lambda) := \sum_{S \subseteq V \text{independent}} \lambda^{|S|} = \sum_{k \geq 0} \alpha_k \lambda^k,$$

where $\alpha_k = \alpha_k(G)$ is the number of independent sets of size k in G. This polynomial encodes a lot of information about the (sizes) of independent sets in G. For example it is easy to see that $Z_G(1)$ gives the number of independent sets in G and $Z'_G(1)/Z_G(1)$ gives the average size of an independent set in G. Knowing the value of $Z_G(\lambda)$ for very large λ would allow one to extract the degree of the polynomial i.e. the size of the largest independent set (which is known to be NP-hard to compute and even to approximate within a constant factor). This already tells us we should not expect to be able to efficiently approximate the independence polynomial at all values of λ.

In this article, we describe a recent technique, the so-called Taylor polynomial interpolation method of Barvinok (first introduced in [7]), for designing approximation algorithms for computational counting problems. Our aim here is to introduce the reader to the ideas behind the method and to give a flavour of the mathematics involved. We do not intend to give a complete survey of results that use the technique and nor do we fully formalise all of the ideas we present. For the latter, we refer the reader to the excellent book of Barvinok [6] and to [72].

One distinguishing feature of the Taylor interpolation method is that, as well as its applications to ordinary counting problems, it also applies to evaluations of generating

[1] As a contrasting example the number of triangles of a graph is polynomial in the size of the graph and can be enumerated by brute force in polynomial time. Of course it is interesting to know whether there is an algorithm for counting triangles that is better than using brute force, but we do not pursue this here.

functions at negative and complex numbers. This is in contrast to earlier techniques. One motivation for understanding such complex evaluations is in quantum computing [90, 26, 70, 49], although we will not discuss this here. Another is that complex evaluations are sometimes useful for real counting problems (see e.g. [3]), and perhaps most importantly, broadening our perspective to the complex plane gives a deeper understanding of the underlying computational complexity of various counting problems (see Section 5 for more discussion on this).

Other techniques for designing approximate counting algorithms (which we will not discuss) include the Markov chain Monte Carlo method (see Jerrum [62] for an excellent introduction to the area) as well as the correlation decay method first introduced by Weitz [89] and Bandyopadhyay and Gamarnik [4] (see e.g. Chapters 5 and 6 of [6] for an introduction). A very recent technique, closely related to Barvinok's interpolation method, is based on the cluster expansion from statistical physics and has been introduced by Jenssen, Keevash and Perkins [61]. We say a few words about this at the end of Section 4.

1.1 Connection to statistical physics

The generating functions for the counting problems we encounter are often studied in the statistical physics community (using different terminology). For example the independence polynomial is known as the partition function of the hard-core model in statistical physics. The hard-core model is a model for gases. Given a closed container of a gas at equilibrium consider examining the gas in a small region of space inside the container. The (discretised) space in the region is represented by a grid graph, where vertices of the graph represent points in space. Each such point can either be occupied or unoccupied by a gas molecule but adjacent points in space cannot both be occupied due to repulsive forces between the molecules. Therefore, at any moment in time, the gas molecules can only occupy an independent set in the grid. The probability $\mathbb{P}(S)$ that at any moment in time the occupied points form a particular independent set S of the grid is proportional to $\lambda^{|S|}$, where $\lambda \in [0, \infty)$ is a temperature-like parameter often called the *fugacity*. A high temperature corresponds to a small value of λ, which, as we intuitively expect, makes it less likely that we see a large set S of occupied points in our small region of space. Since $\mathbb{P}(S) \propto \lambda^{|S|}$, and $\sum_{S \subseteq V \text{ independent}} \mathbb{P}(S) = 1$, we see that $\mathbb{P}(S) = \lambda^{|S|}/Z_G(\lambda)$. Here we see the independence polynomial $Z_G(\lambda)$ (known here as the partition function of the hard-core model) appearing as the normalising constant in the probability. Again, this partition function is much more than just a normalising constant, and encodes a lot of physical information about the system. For example, by considering the limiting behaviour of $\ln Z_G(\lambda)/|V(G)|$ and its derivatives for larger and larger graphs (usually grids), discontinuities of these limit functions give information about phase transitions in the system, that is, sharp changes in the physical parameters associated with the system indicating a qualitative change in the system. We direct the reader to [46] for a comprehensive and rigourous mathematical treatment of phase transitions for many models and to [81, 38, 53] for more on the hard-core model. We will not be concerned directly with the statistical physics, but some results originally proved by statistical physicists will be used in the algorithmic approach we describe.

1.2 Preliminaries

We have already mentioned the independence polynomial as an example of a graph polynomial that we may wish to approximate. The independence polynomial will serve as a running example throughout the article to illustrate various ideas. Here we mention a few basic properties of the independence polynomial to give the reader a feel for this object.

Recall that $Z_G(\lambda) = \sum \lambda^{|S|}$, where the sum is over all independent sets S of G. The first easy but important fact to note is that the empty set is an independent set, so $Z_G(0)$ (i.e. the constant term in the polynomial) is always 1. Another important fact is that the independence polynomial is multiplicative, that is $Z_{G_1 \cup G_2}(\lambda) = Z_{G_1}(\lambda) Z_{G_2}(\lambda)$, where we write $G_1 \cup G_2$ for the disjoint union of the graphs G_1 and G_2. This is because every independent set S of $G_1 \cup G_2$ can be written uniquely as $S = S_1 \cup S_2$, where S_i is an independent set of G_i. Therefore $\lambda^{|S|} = \lambda^{|S_1|}\lambda^{|S_2|}$, which allows us to factorise the sum. Using this multiplicative property, we also see, for example, that the independence polynomial of k isolated vertices is $(1+\lambda)^k$. One can also see directly that the complete graph on k vertices has independence polynomial $1 + k\lambda$.

We now describe the type of algorithm we ideally wish to obtain for our graph counting problems. Suppose $p = p(G)$ is a graph parameter, e.g. $p(G)$ is the number of independent sets in G, or $p(G) = Z_G(\lambda)$ for some fixed λ. Note that we allow $p(G)$ to be a complex number. A *fully polynomial-time approximation scheme* (or FPTAS for short) for p is an algorithm that takes as input a graph G and an error tolerance $\varepsilon > 0$ and outputs a (complex) number N such that $N = e^{\varepsilon t} p(G)$ for some $t \in \mathbb{C}$ with $|t| \leq 1$ in time polynomial in $|G|$ (the number of vertices of G) and ε^{-1}. Note that when ε is small, we have $N = e^{\varepsilon t} p(G) \approx (1 + \varepsilon t) p(G)$, so that N is roughly within a distance $\varepsilon |p(G)|$ of the true value of $p(G)$. For this reason we call such output N a multiplicative ε-approximation (for $p(G)$).[2] We also discuss algorithms that provide the same approximation as above but that run in time super-polynomial in $|G|$.

2 Barvinok's interpolation method

In this section we describe the Taylor polynomial interpolation method of Barvinok, a method that can be applied to a wide variety of counting problems. Consider some graph polynomial, that is, each graph G has some associated polynomial $P(z) = P_G(z)$. As with the independence polynomial, we should imagine that P_G is not directly accessible, i.e. at least some of its coefficients are difficult to compute from G. We will

[2] Note that this definition of FPTAS is consistent with the usual notion of FPTAS for real parameters.

however assume that the degree of the polynomial P_G is always bounded by a constant times $|G|$; this is certainly the case for the independence polynomial and is easy to verify for most graph polynomials one might consider. Our goal is to (efficiently) obtain a multiplicative ε-approximation for $P_G(z)$ for $z \in \mathbb{C}$.

The insight of Barvinok was to use Taylor's theorem, about power series approximations of smooth functions, to obtain the desired approximation. At first sight we seem to gain nothing from Taylor's theorem because the Taylor series of a polynomial is simply the polynomial itself. However, notice that the truncated Taylor series of a (non-polynomial) function gives an *additive* ε-approximation to the function, whereas we are interested in a *multiplicative* ε-approximation. Therefore, rather than considering the Taylor series of $P_G(z)$, we should in fact consider the Taylor series of $g(z) := \ln P_G(z)$ and then take the exponential of the result to obtain the desired approximation.[3]

To this end, consider the Taylor series of $g(z)$ about zero:

$$g(z) = \sum_{k=0}^{\infty} \frac{g^{(k)}(0)}{k!} z^k,$$

where $g^{(k)}$ denotes the kth derivative of g. Unfortunately, the Taylor series of a function does not usually converge for all $z \in \mathbb{C}$. We will return shortly to the question of convergence, but let us assume for now that the Taylor series does converge to $g(z)$ for a value of z we are interested in. In that case, if we write $T_m(z)$ for the first m terms of the Taylor series of g above, then for m sufficiently large, we will have that $|T_m(z) - g(z)| < \varepsilon$, i.e. $T_m(z) = g(z) + \varepsilon t$ for some $t \in \mathbb{C}$ with $|t| < 1$. Taking the exponential of both sides of the equation, we obtain $\exp(T_m(z)) = \exp(\varepsilon t) P_G(z)$ i.e. $\exp(T_m(z))$ a multiplicative ε-approximation for $P_G(z)$.

This gives us the desired approximation, but several questions remain. Firstly, there is the question of convergence mentioned above. Secondly, if the Taylor series does converge, then how large does m have to be to guarantee that $|T_m(z) - g(z)| < \varepsilon$? Finally, how can we actually compute $T_m(z)$ in order to compute our approximation $\exp(T_m(z))$ for $P_G(z)$? Note that we do not have direct access to the numbers $g^{(k)}(0)$; these have to be computed in some way.

For the first question of convergence, Taylor's theorem says that the Taylor series for g converges inside the disk $D_R := \{z \in \mathbb{C} : |z| \leq R\}$ for any $R > 0$ provided that g is analytic inside D_R. In our case, this holds provided $P_G(z) \neq 0$ for all $z \in D_R$.[4] So the Taylor series will converge inside the largest disk that contains no roots of $P_G(z)$. Establishing such zero-freeness results for particular graph polynomials will be the subject of Section 4.

The second question concerns the rate of convergence of the Taylor series of g. Here we take advantage of the particular form of g as the logarithm of a polynomial. If

[3] In order for $g(z)$ to be well-defined we need to fix a branch of the logarithm here; we say more below.

[4] Formally, to ensure g is analytic, we fix $\ln P_G(0)$, and take the branch of $g(z) = \ln P_G(z)$ on D_R given by $g(z) = \ln P_G(0) + \int_0^z P_G'(w)/P_G(w)dw$.

$\eta_1, \ldots, \eta_d$ are the (complex) roots[5] of $P_G(z)$ then we can write $P_G(z) = a\prod_{i=1}^{d}(1 - \frac{z}{\eta_i})$, where $a = P_G(0)$ is assumed to be non-zero. Then taking logarithms of both sides, we have

$$g(z) = \ln(a) + \sum_{i=1}^{d} \ln(1 - (z/\eta_i)).$$

Using that the Taylor series of $\ln(1-z) = -z - \frac{z^2}{2} - \frac{z^3}{3} - \cdots$ for $|z| < 1$, we obtain the Taylor series of g as

$$g(z) = \ln(a) - \sum_{i=1}^{d}\sum_{k=1}^{\infty} \frac{(z/\eta_i)^k}{k}$$

for $|z| < \min_i |\eta_i|$ (precisely the condition of zero-freeness mentioned above). Assuming $|z| \leq \delta \min_i |\eta_i|$ for some $\delta \in (0,1)$, this gives

$$|g(z) - T_m(z)| \leq \sum_{i=1}^{d}\sum_{k=m}^{\infty} \left|\frac{(z/\eta_i)^k}{k}\right| \leq \sum_{i=1}^{d}\sum_{k=m}^{\infty} \delta^k = \frac{d\delta^m}{1-\delta}.$$

In order to bound the last expression by ε, it is sufficient to take $m \geq C\ln(d/\varepsilon)$ for some constant C depending on δ. For such m we have that $\exp(T_m(z))$ is a multiplicative ε-approximation for $P_G(z)$.

The final question of actually computing $T_m(z)$ is more subtle and will only be partially addressed here and in the next section. We will show that if we know the values of the first $m = C\ln(d/\varepsilon)$ coefficients of $P_G(z)$ then we can compute the derivatives $g^{(0)}, g^{(1)}, \ldots, g^{(m)}$ in time poly(m). However, we do not typically have immediate access to the first m coefficients of our graph polynomials. For example, in the case of the independence polynomial $Z_G(\lambda)$, the coefficient α_k of λ^k is the number of independent sets of size k in G: computing this naively with a brute force approach of checking every k-tuple of vertices takes time n^k (where $n = |G|$) and so computing α_m takes time $n^m = n^{O(\ln n)}$ (noting that the degree of Z_G i.e. the size of the largest independent set could be and often is linear in n). In the next section, we show how to compute $\alpha_0, \ldots, \alpha_m$ in poly(n) time and the idea turns out to generalise for many other graph polynomials of interest. For now, here is how to compute $T_m(z)$ given the first m coefficients of $P_G(z)$.

Suppose $P(z) = P_G(z) = a_0 + a_1 z + \cdots + a_d z^d$. We defined $g(z) = \ln P_G(z)$. We know $g^{(0)}(0) = g(0) = \ln(a_0)$. If we differentiate once and rearrange, we obtain $g^{(1)}(z)P(z) = P^{(1)}(z)$. If we now repeatedly differentiate this expression, we obtain the

[5] Again, we do not typically have access to the roots of P_G; we work with the roots only in the analysis of the algorithm.

following expressions:

$$\begin{aligned}
P^{(1)} &= g^{(1)}P^{(0)}\\
P^{(2)} &= g^{(2)}P^{(0)} + g^{(1)}P^{(1)}\\
&\vdots\\
P^{(r)} &= g^{(r)}P^{(0)} + \binom{r-1}{1}g^{(r-1)}P^{(1)} + \binom{r-1}{2}g^{(r-2)}P^{(2)} + \cdots + \binom{r-1}{r-1}g^{(1)}P^{(r-1)}.
\end{aligned}$$

Evaluating these expressions at zero, and noting that $P^{(r)}(0) = r!a_r$, we obtain

$$\begin{aligned}
a_1 &= a_0 g^{(1)}(0)\\
2a_2 &= a_0 g^{(2)}(0) + a_1 g^{(1)}(0)\\
&\vdots\\
ra_r &= a_0 g^{(r)}(0) + \frac{(r-1)!}{(r-1)!}a_1 g^{(r-1)}(0) + \frac{(r-1)!}{(r-2)!}a_2 g^{(r-2)}(0) + \cdots + \frac{(r-1)!}{1!}a_{r-1}g^{(1)}(0).
\end{aligned}$$

We see that if we know $a_0, \dots, a_r$ and we have computed $g^{(0)}(0), \dots, g^{(r-1)}(0)$, then we can use the rth equation above to compute $g^{(r)}(0)$ in time $O(r)$. Therefore given $a_0, \dots, a_m$, we can compute $T_m(z)$ in $O(m^2)$ time.

The following summarises what we have shown in this section and is the essence of the Taylor polynomial interpolation method.

Theorem 2.1. *Suppose $\mathcal{G}$ is an (infinite) set of graphs and for each $G \in \mathcal{G}$, $P_G(z)$ is a polynomial associated with G, where*

$$P_G(z) = \sum_{i=0}^{d(G)} a_i(G)z^i$$

Suppose there exists $R > 0$ and a function $T : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ with the properties that

(i) $P_G(z) \neq 0$ whenever $|z| \leq R$ for all graphs $G \in \mathcal{G}$, and

(ii) we are able to compute $a_i(G)$ in time bounded by $T(|G|, i)$, where we assume for convenience that T is non-decreasing in both arguments.

Then there is an algorithm, which, given input $G \in \mathcal{G}$, $\varepsilon > 0$, and $z \in \mathbb{C}$ with $|z| < R$, computes a multiplicative ε-approximation of $P_G(z)$ in time $mT(n, m) + O(m^2)$, where $n = |G|$ and $m := C\ln(d(G)/\varepsilon)$ (as defined earlier).

Some remarks are in order. The theorem is formulated for a general class of graphs $\mathcal{G}$ rather than all graphs because often, we are only able to establish conditions (i) and (ii) effectively for certain types of graphs (typically bounded degree graphs). This is best illustrated by applying the result above to the independence polynomial.

For the independence polynomial, if we consider $\mathcal{G}$ to be the set of all graphs, then there is no zero-free disk of positive radius[6] so we can only take $R = 0$ in condition (i). However, if we restrict $\mathcal{G}$ to graphs of maximum degree Δ, we will see in Section 4 that we can take $R = (\Delta - 1)^{\Delta-1}/\Delta^{\Delta}$. Similarly for condition (ii), if we take $\mathcal{G}$ to be all graphs, then the brute force approach mentioned earlier is essentially the only way to compute coefficients of $Z_G(\lambda)$ giving $T(n,k) = O(n^k)$: overall this gives a super-polynomial running time of $mT(n,m) + O(m^2) = n^{O(m)} + O(m^2) = n^{O(\ln(n/\varepsilon))}$. Such a quasi-polynomial running time is already quite promising because it is significantly better than the exponential running time of a brute-force algorithm. However, in the next section we will see that for graphs of maximum degree at most Δ, we can compute the coefficients much faster and take $T(n,k) = \text{poly}(n)\Delta^{O(k)}$, thereby establishing an overall polynomial running time of $T(n,m) = \text{poly}(n)\Delta^{O(m)} = \text{poly}(n)\Delta^{O(\ln(n/\varepsilon))} = (n/\varepsilon)^{O(\ln\Delta)}$. Combining the results from the next two sections will therefore give an FPTAS for computing $Z_G(\lambda)$ on graphs of maximum degree at most Δ provided $|\lambda| < (\Delta - 1)^{\Delta-1}/\Delta^{\Delta}$.

Finally, we remark that one can in fact relax condition (i) to include regions that are not necessarily disks provided the region is "thick" in a certain sense and contains the point 0. Concretely one should think of a small neighbourhood of a real interval or a sector region. Relaxing condition (i) to non-disk regions is achieved by making suitable polynomial transformations of P_G; see Section 2.2.2 of [6] and [10] for details.

3 Polynomial running time for bounded degree graphs

In the last section we saw how we can use Taylor's theorem to design algorithms to approximate graph polynomials. Let $P = P_G$ be a graph polynomial. Examining Theorem 2.1, we require two ingredients to establish an approximation algorithm to compute $P_G(z)$. First we need to establish a zero-free disk for P_G; this will be discussed in detail in the next section. Second, we need to be able to efficiently compute the first $O(\ln|G|)$ coefficients of P_G, which we discuss in detail here. There is usually a straightforward, direct approach for computing these coefficients, which leads to quasi-polynomial time algorithms, but which is not fast enough for an FPTAS. We already saw this in the last section with the independence polynomial, where we saw that computing the coefficients naively leads to an $n^{O(\ln n)}$-time algorithm.[7] In this section, we show how to compute the coefficients of the independence polynomial more efficiently for *bounded degree graphs*. The technique generalises to many other graph polynomials but all the key ideas are best understood through the concrete example of the independence polynomial. We give the statement for general graph

[6] The k-vertex complete graph has independence polynomial $1 + k\lambda$, so its roots tend to 0.

[7] Computing the coefficients naively often gives us quasi-polynomial time approximation algorithms as with the example of the independence polynomial. This is already very good because it is a significant improvement on the exponential time taken to enumerate independent sets in a graph. Achieving the polynomial runtime of an FPTAS is considered to be the gold standard in the area.

polynomials at the end of the section.

It is worth noting that, barring a few exceptions, the setting of bounded degree graphs is often the setting of interest. For example, the problem of computing a multiplicative ε-approximation for $Z_G(z)$ is known to be computationally hard for all complex $z \neq 0$ if we have no restriction on G; see [84, 51, 23].

3.1 Computing the coefficients of the independence polynomial efficiently

Recall that the independence polynomial $Z_G(\lambda)$ is given by

$$Z_G(\lambda) = \sum_{k \geq 0} \alpha_k \lambda^k,$$

where $\alpha_k = \alpha_k(G)$ is the number of independent sets of size k in G. Throughout this section we focus on bounded degree graphs and write $\mathcal{G}_\Delta$ for the set of graphs of maximum degree at most Δ. If we apply Theorem 2.1 to $Z_G(\lambda)$ (with $G \in \mathcal{G}_\Delta$), assuming we have some suitable zero-free disk containing λ, Theorem 2.1 gives us an algorithm to compute a multiplicative ε-approximation of $Z_G(\lambda)$ in time $mT(n,m) + O(m^2)$, where $T(n,i)$ is the time needed to compute $\alpha_i(G)$ for n-vertex graphs $G \in \mathcal{G}_\Delta$ and $m \leq C \ln(n/\varepsilon)$. We will sketch a proof of the following.

Theorem 3.1. *For $G \in \mathcal{G}_\Delta$, we can compute $\alpha_i(G)$ in time $\mathrm{poly}(|G|)\Delta^{O(i)}$, i.e. for n-vertex graphs of maximum degree at most Δ, we can take $T(n,i) = \mathrm{poly}(n)\Delta^{O(i)}$.*

Using the theorem above, Theorem 2.1 gives us an approximation algorithm for the independence polynomial with running time

$$mT(n,m) + O(m^2) = \mathrm{poly(n)}\Delta^{O(\ln(n/\varepsilon)} = \mathrm{poly(n)}(n/\varepsilon)^{O(\ln(\Delta))}.$$

We see that this running time is of the form required for a fully polynomial time approximation scheme. We now sketch the proof of Theorem 3.1.

Sketch proof of Theorem 3.1

We begin by generalising the algorithmic problem we are interested in. We are interested in computing $\alpha_k(G)$, the number of independent sets of size k in G, when $G \in \mathcal{G}_\Delta$. Equivalently, $\alpha_k(G)$ is the number of induced copies of the graph I_k in G, where I_k is the graph consisting of k vertices and no edges. Generally for graphs H and G, write $\mathrm{ind}(H,G)$ for the number of induced copies[8] of H in G. Then $\alpha_k(G) = \mathrm{ind}(I_k, G)$.

The first observation is that, while we do not know how to efficiently compute $\mathrm{ind}(I_k, G)$, it is not too hard to efficiently compute $\mathrm{ind}(H,G)$, when H is connected.

[8] The number of induced copies of a graph H in the graph $G = (V,E)$ is defined as the number of vertex subsets $S \subseteq V$ such that $G[S] = H$.

Observation 3.1. We can compute $\mathrm{ind}(H,G)$ in time $\mathrm{poly}(n)\Delta^{O(k)}$, where $G \in \mathcal{G}_\Delta$, $n = |G|$ and $k = |H|$.

To see this, we first pick any spanning tree T in H (i.e. a subgraph of H that is a tree and that uses all k vertices of H). Such a spanning tree exists because H is connected. The idea is to find all (not necessarily induced) copies of T in G and to check which of the copies of T extend to an induced copy of H. This accounts for all induced copies of H because every induced copy of H in G contains a (not necessarily induced) copy of T in G.

There are only relatively few (not necessarily induced) copies of T in G. Indeed, first we enumerate the vertices of T in a breadth-first ordering $v_1, v_2, \ldots, v_k$. We embed T into G one vertex at a time in order. There are n choices of where to embed v_1. Each subsequent vertex of T has at most Δ possibilities for its embedding into G because when we come to embed v_i, its parent in T (say $v_{i'}$) has already been embedded as some vertex $x_{i'}$ in G, so the embedding of v_i must be a neighbour of $x_{i'}$ in G. Therefore altogether there are at most $n\Delta^{k-1}$ embeddings of T in G and each such embedding of T is checked to see if it gives an induced copy of of H.

From Observation 3.1, we see that we can compute $\mathrm{ind}(H,G)$, when H is connected, but the graph H we are interested in, namely I_k, is very much disconnected. It would be useful if we could express $\mathrm{ind}(I_k,G)$ in terms of $\mathrm{ind}(H,G)$ for connected H. A trivial case of this is the fact that $\mathrm{ind}(I_2,G) = \binom{n}{2} - \mathrm{ind}(e,G)$, where e is the graph on two vertices with an edge between them. This says nothing other than that the number of edges and non-edges in an n-vertex graph sum to $\binom{n}{2}$. With a little more work, we can express $\mathrm{ind}(I_3,G)$ in terms of induced counts of connected graphs as follows. There are four graphs on three vertices, namely I_3, the triangle denoted T, the path on three vertices denoted P_3 and the disjoint union of an edge and a vertex denoted $e + I_1$. By enumerating all induced subgraphs of G on three vertices, we have

$$\mathrm{ind}(I_3,G) = \binom{n}{3} - \mathrm{ind}(T,G) - \mathrm{ind}(P_2,G) - \mathrm{ind}(e+I_1,G).$$

The only disconnected graph on the right hand side is $e + I_1$, and by simple counting, it is not too hard to show that

$$\mathrm{ind}(e+I_1,G) = (n-2)\mathrm{ind}(e,G) - 2\mathrm{ind}(P_3,G) - 3\mathrm{ind}(T,G).$$

Substituting the second formula into the first gives an expression for $\mathrm{ind}(I_3,G)$ in terms of induced counts of connected graphs.

These calculations suggest that it is possible to express $\mathrm{ind}(I_k,G)$ in terms of induced counts $\mathrm{ind}(H,G)$ for connected graphs H, but that the calculations and formulae will get cumbersome. A new idea is needed to approach the problem in a systematic and manageable way. The next observation is the key insight to overcoming this hurdle and is at the heart of the proof of Theorem 3.1. It was proved by Csikvári and Frenkel [36]; the proof is short and can also be found in [72].

Observation 3.2. Suppose $\tau(G)$ is an additive graph property, meaning that it satisfies the following two properties.

(i) $\tau(G)$ can be written as sum of products of induced graph counts, i.e. for all G

$$\tau(G) = \sum_{i=1}^{r} \mu_i \prod_{H \in \mathcal{H}_i} \mathrm{ind}(H, G),$$

where $\mathcal{H}_i$ is a (finite) set of graphs and $\mu_i \in \mathbb{C}$ is a constant for each $i = 1, \dots, r$, and

(ii) $\tau(G_1 \cup G_2) = \tau(G_1) + \tau(G_2)$ for all graphs G_1 and G_2.

Then τ is in fact of a simpler form, namely, for all graphs G, we have

$$\tau(G) = \lambda_1 \mathrm{ind}(H_1, G) + \cdots + \lambda_s \mathrm{ind}(H_s, G),$$

where $H_1, \dots, H_s$ are *connected* graphs and $\lambda_1, \dots, \lambda_s \in \mathbb{C}$.

The observation above says that every additive graph parameter is a linear combination of $\mathrm{ind}(H_i, G)$ for *connected* H_i, and so by Observation 3.1, such additive graph parameters can be computed efficiently.[9] Our task now is reduced to the task of expressing $\alpha_k(G) = \mathrm{ind}(I_k, G)$ in terms of additive graph parameters. In order to do this, we now switch from the combinatorial to the polynomial perspective of $\alpha_k(G)$.

Recall that the $\alpha_k(G)$ are the coefficients of the independence polynomial Z_G, i.e. $Z_G(\lambda) = \alpha_0 + \alpha_1\lambda + \cdots \alpha_d\lambda^d$. Suppose that $\eta_1, \dots, \eta_d$ are the roots of Z_G. Noting that the constant term α_0 is one, we can write $Z_G(\lambda) = (1 - \eta_1^{-1}\lambda) \cdots (1 - \eta_d^{-1}\lambda)$. While we cannot compute the η_i directly, we can relate them to the coefficients α_k by expanding the product above. We see that the α_k are the elementary symmetric polynomials in η_i^{-1}, namely

$$\alpha_0 = 1, \qquad \alpha_1 = -\sum_{1 \le i \le d} \eta_i^{-1}, \qquad \alpha_2 = \sum_{1 \le i < j \le d} \eta_i^{-1}\eta_j^{-1} \qquad \text{etc.}$$

Another important class of symmetric polynomials are the power sums. Let us define the ith power sum p_i to be

$$p_i = \eta_1^{-i} + \cdots + \eta_d^{-i}.$$

It is well known that the power sums can be related to the elementary symmetric polynomials using the Newton identities. There are several short derivations of these identities. In the context of our problem, the Newton identities give the following

[9] Actually efficient computation is not immediate because it depends on the number and size of the H_i; we address this later.

expressions relating the α_i and the p_i.

$$\begin{aligned}
-\alpha_1 &= \alpha_0 p_1 \\
-2\alpha_2 &= \alpha_0 p_2 + \alpha_1 p_1 \\
-3\alpha_3 &= \alpha_0 p_3 + \alpha_1 p_2 + \alpha_2 p_1 \\
&\vdots \\
-t\alpha_t &= \alpha_0 p_t + \alpha_1 p_{t-1} + \cdots + \alpha_{t-1} p_1.
\end{aligned}$$

From this it is easy to see that if we know the values of the p_i then we can inductively compute the α_i. Indeed, if we know the values of $p_1, \ldots, p_t$, and we also know (by induction) the values of $\alpha_1 \ldots, \alpha_{t-1}$ then using the t^{th} identity, we can compute α_t. Thus the problem of efficiently computing the α_i is reduced to that of efficiently computing the p_i. It is possible to efficiently compute the power sums because, as the reader may have guessed, the power sums are additive graph parameters.

Observation 3.3. The power sums $p_i = p_i(G)$ as defined above have the property of being additive graph parameters.

It is easy to verify that p_i satisfies the second property of an additive graph parameter, namely that $p_i(G_1 \cup G_2) = p_i(G_1) + p_i(G_2)$ for any graphs G_1 and G_2. Indeed, since $Z_{G_1 \cup G_2} = Z_{G_1} Z_{G_2}$ (see Section 1.2), if $\eta_1, \ldots, \eta_d$ are the roots of of Z_{G_1} and $\nu_1, \ldots, \nu_{d'}$ are the roots of Z_{G_2} then $\eta_1, \ldots, \eta_d, \nu_1, \ldots, \nu_{d'}$ are the roots of $Z_{G_1 \cup G_2}$ so that

$$p_i(G_1 \cup G_2) = \eta_1^{-i} + \cdots + \eta_d^{-i} + \nu_1^{-i} + \cdots + \nu_{d'}^{-i} = p_i(G_1) + p_i(G_2).$$

For the first property, we use the Newton identities. Note that, since $\alpha_0 = 1$, we can rearrange the t^{th} identity and express p_t as a sum of products of $p_1, \ldots, p_{t-1}$ and $\alpha_1, \ldots, \alpha_t$. We know that the α_i are induced graph counts, and if we assume by induction that $p_1, \ldots, p_{t-1}$ are also sums of products of induced graph counts, then we see that p_t is also a sum of products of induced graph counts and so satisfies property (i) of an additive graph parameter.

We now have all the ingredients to explain how to compute the α_k efficiently. We can compute the power sums p_i efficiently. This is because the power sums are additive graph parameters (Observation 3.3) and they are therefore linear combinations of induced counts of *connected* graphs (Observation 3.2). Each induced graph count $\text{ind}(H, G)$ in this linear combination can be computed efficiently when G is of bounded degree since H is connected (Observation 3.1) thus allowing us to compute the power sums efficiently. Once we have computed the power sums $p_1, p_2, \ldots$, we can inductively compute the α_i using the Newton identities.

This gives the main ideas of the argument although there are a few subtleties that we have glossed over. The main one is that it is not quite obvious that we can compute the power sums $p_i(G)$ efficiently, i.e. in time $\text{poly}(|\text{G}|)\Delta^{O(i)}$. While the $p_i(G)$ can be expressed as a linear combination of induced counts of connected graphs

$$p_i(G) = \lambda_1 \text{ind}(H_1, G) + \cdots + \lambda_s \text{ind}(H_s, G),$$

we have not said how to find $H_1, \ldots, H_s$ and $\lambda_1, \ldots, \lambda_s$. Conceivably, s could be superexponential in i or the H_i could have size superlinear in i; in either case we would not automatically get the desired running time. However, by using the Newton identities more carefully, and using the fact that G has bounded degree it is not too difficult to overcome these technical obstacles. All the details can be found in [72].

3.2 Computing the coefficients of other graph polynomials efficiently

In Section 3.1, we described the main idea of how we can efficiently compute the first $\ln |G|$ coefficients of the independence polynomial Z_G for graphs G of bounded degree. The ideas can be generalised to work for many other graph polynomials of interest.

What are the crucial properties of the independence polynomial Z_G that we use in the sketch proof of Theorem 3.1? The whole proof is based around manipulating induced graph counts, so we certainly need the coefficients of Z_G to be (functions of) induced graph counts. We also crucially need that Z_G is multiplicative, which allows us to conclude that the power sums are additive, therefore allowing us to compute them efficiently.

In [72], we show that if a graph polynomial $P = P_G$ satisfies certain properties given below, then its coefficients can be computed efficiently for bounded degree graphs i.e. the ith coefficient of P_G can be computed in time $\text{poly}(n)\Delta^{O(i)}$ where G is an n-vertex graph of maximum degree at most Δ. As with the independence polynomial, this is enough to use the Taylor polynomial interpolation method from Section 2 to give an approximation algorithm for computing $P_G(z)$ (provided z is in a suitable zero-free disk) with the required run time of an FPTAS.

Suppose $P = P_G$ is a graph polynomial given by $P_G(z) = a_0 + a_1 z + \cdots + a_d z^d$. Suppose that P satisfies the following properties for some fixed constant $\alpha > 0$:

(i) for each ℓ, the ℓth coefficient of P can be expressed as a "α-bounded" linear combination of induced graph counts, that is, for all $G \in \mathcal{G}_\Delta$

$$a_\ell(G) = \sum_H \zeta_{H,\ell} \text{ind}(H, G),$$

where the sum is over graphs H with at most $\alpha\ell$ vertices and $\zeta_{H,\ell} \in \mathbb{C}$ are constants (independent of G);

(ii) in property (i), for each H we can compute $\zeta_{H,\ell}$ in time $\exp(O(|H|)$; and

(iii) P_G is multiplicative, i.e. $P_{G_1 \cup G_2} = P_{G_1} P_{G_2}$.

Then we can compute $a_i(G)$ in time $\text{poly}(|G|)\Delta^{O(i)}$. Again, using the Taylor polynomial interpolation method, this leads to an FPTAS for approximating $P_G(z)$ for $G \in \mathcal{G}_\Delta$, again provided we establish a suitable zero-free disk containing z.

Note that in the case of the independence polynomial, properties (i) and (ii) are trivial and we saw it is easy to verify property (iii). These properties also hold for various other graph polynomials including the matching polynomial, the chromatic polynomial, and the Tutte polynomial.[10] We will not check these here, but refer the interested reader to [72]. It is also worth noting that the technique described in this section can be adapted and applied to polynomials beyond those satisfying properties (i)-(iii) above; see [68, 15, 70].

In Section 2 we explained how one can design algorithms for approximating graph polynomials using Taylor's theorem. In this section, we showed how to make these algorithms efficient (having the running time of an FPTAS) for many graph polynomials provided we restrict attention to bounded degree graphs. We have seen in Section 2 that essential to all of these algorithms is to establish a suitable zero-free disk or zero-free region in the complex plane for the graph polynomial in question. Our discussion of algorithms ends at this point and in the next section, we turn our attention entirely to the independent problem of establishing these zero-free regions.

4 Techniques for proving absence of zeros

In the previous sections, we have sketched how the problem of approximately evaluating graph polynomials (particularly the independence polynomial) in a region of the complex plane is reduced to the problem of establishing that the polynomial has no zeros in that region. There is a long history of proving such results about the locations of zeros of graph polynomials and partition functions. The techniques used often have their origin in statistical physics but have now been picked up and extended by the theoretical computer science community. In this section we will discuss three different techniques.

4.1 Recursion and ratios

Many graph polynomials satisfy recursions in which the polynomial for a given graph can be expressed in terms of the polynomial for smaller graphs. Such recursions allow us to prove properties about the graph polynomial, such as absence of zeros, by induction. However, rather than working with the polynomials directly, it is often more productive to work instead with related quantities. We illustrate this approach through our running example of the independence polynomial and at the end of the section we direct the reader to further work in which this technique is employed.

Our aim is to sketch a proof of the following result due to Shearer [83], Dobrushin [38] and Scott and Sokal [81]:

Theorem 4.1. *Let $G = (V, E)$ be a graph with maximum degree $\Delta \geq 2$ and let $\lambda \in \mathbb{C}$ satisfy $|\lambda| \leq \lambda^*(\Delta) := \frac{(\Delta-1)^{\Delta-1}}{\Delta^\Delta}$. Then $Z_G(\lambda) \neq 0$.*

[10] The Tutte polynomial is a polynomial in two variables, but the properties above hold if one of the variables is fixed

Let us briefly discuss this result before delving into the proof. First, recall that by the Taylor polynomial interpolation method (particularly Theorem 2.1 and Theorem 3.1), this result immediately implies an FPTAS for computing $Z_G(\lambda)$ for $G \in \mathcal{G}_\Delta$ inside the zero-free disk given by $|\lambda| < \lambda^*(\Delta)$. Second, note that if we are only interested in zero-free *disks*, then one cannot improve Theorem 4.1 in the sense that we cannot increase the constant $\lambda^*(\Delta)$. Indeed, one can show that there is a sequence of graphs G_n (in fact trees) of maximum degree Δ and negative numbers λ_n such that $Z_{G_n}(\lambda_n) = 0$ and $\lambda_n \to -\lambda^*(\Delta)$ [81]. However there has been a lot of interest recently in establishing zero-freeness for non-disk regions. Most notably, it was shown recently [74] that $Z_G(\lambda) \neq 0$ whenever $G \in \mathcal{G}_\Delta$ and $\lambda \in R \subseteq \mathbb{C}$ where R is an open set containing the interval $[0, \lambda_c(\Delta))$ and $\lambda_c(\Delta) := (\Delta-1)^{\Delta-1}/(\Delta-2)^\Delta$. One significance of $\lambda_c(\Delta)$ is that it is an algorithmic threshold for real parameters λ: using the interpolation method, the result in [74] implies that there is an FPTAS[11] to compute $Z_G(\lambda)$ whenever $G \in \mathcal{G}_\Delta$ and $\lambda \in [0, \lambda_c(\Delta))$, while for $\lambda > \lambda_c(\Delta)$, it is known that there is no such FPTAS unless $P = NP$ [84, 51].

We now discuss the proof of Theorem 4.1. Let $G = (V, E)$ be a graph and fix a vertex $v \in V$. We can write down a recursion for $Z_G(\lambda) = \sum_{S \subseteq V \text{ independent}} \lambda^{|S|}$ by splitting the sum over those independent sets that do not contain v and those that do to obtain

$$Z_G(\lambda) = Z_{G-v}(\lambda) + \lambda Z_{G\setminus[N[v]}(\lambda), \tag{1}$$

where $G - v$ (resp. $G \setminus N[v]$) denote the graphs obtained from G by removing v (resp. v and its neighbours in G). As mentioned earlier, rather than working directly with a recursion for Z_G, it turns out to be more useful to work with a recursion of a related quantity. Define the *ratio*, $R_{G,v}$, by

$$R_{G,v}(\lambda) := \frac{\lambda Z_{G\setminus N[v]}(\lambda)}{Z_{G-v}(\lambda)}. \tag{2}$$

Observe that provided $Z_{G-v}(\lambda) \neq 0$, we have $Z_G(\lambda) = 0$ if and only if $R_{G,v}(\lambda) = -1$ (using (1)). So to prove absence of zeros it suffices to inductively show that the ratios avoid -1.

Next we establish a recursion for these ratios. Let G be a graph with fixed vertex u_0 and let $\lambda \in \mathbb{C}$. Let $u_1, \ldots, u_d$ be the neighbours of u_0 in G (in any order). Set $G_0 = G - u_0$ and define for $i = 1, \ldots, d$, $G_i := G_{i-1} - u_i$ (so $G_d = G \setminus N[u_0]$). Suppose that $Z_{G_i}(\lambda) \neq 0$ for all $i = 0, \ldots, d$. Then we use 'telescoping' to write

$$\frac{R_{G,u_0}(\lambda)}{\lambda} = \frac{Z_{G_d}(\lambda)}{Z_{G_0}(\lambda)} = \frac{Z_{G_1}(\lambda)}{Z_{G_0}(\lambda)} \cdot \frac{Z_{G_2}(\lambda)}{Z_{G_1}(\lambda)} \cdots \frac{Z_{G_d}(\lambda)}{Z_{G_{d-1}}(\lambda)}.$$

Applying (1) to each of the denominators and after some rearranging we end up with the following identity:

$$R_{G,u_0}(\lambda) = \frac{\lambda}{\prod_{i=1}^d (1 + R_{G_{i-1},u_i}(\lambda))}. \tag{3}$$

[11] In fact, an FPTAS was established earlier in [89] using the correlation decay method.

The identity above captures all the relevant combinatorics of independent sets that we need and the rest of the proof essentially boils down to proving a property about the above recursion.

Proof of Theorem 4.1. We may assume that G is connected (if G has connected components $H_1, \ldots, H_k$ then $Z_G(\lambda) = Z_{H_1}(\lambda) \cdots Z_{H_k}(\lambda)$ and so it is sufficient to prove the theorem for each H_i).

Fix $v_0 \in V$. We will show by induction that the following holds for all $U \subseteq V \setminus \{v_0\}$:

(i) $Z_{G[U]}(\lambda) \neq 0$,

(ii) if $u_0 \in U$ has a neighbour in $V \setminus U$, then $|R_{G[U],u_0}(\lambda)| < 1/\Delta$.

Indeed if $|U| = 0$ then this is trivially true, so suppose that $|U| > 0$. Then since G is connected, there is $u_0 \in U$ that has a neighbour $v \in V \setminus U$. Let us write $H = G[U]$ and let $u_1, \ldots, u_d$ be the neighbours of u_0 in H. Let $H_0 = H - u_0$ and $H_i = H_{i-1} - u_i$ for $i > 0$. Then, by induction $Z_{H_i}(\lambda) \neq 0$ and $|R_{H_i,u_{i+1}}(\lambda)| < 1/\Delta$ (since u_{i+1} has a neighbour in $U \setminus V(H_i)$, namely u_0). So we may use (3) to conclude that

$$|R_{H,u_0}(\lambda)| = \frac{|\lambda|}{\prod_{i=1}^{d} |1 + R_{H_{i-1},u_i}(\lambda)|} < |\lambda|(1 - 1/\Delta)^{-d}$$
$$\leq |\lambda| \left(\frac{\Delta - 1}{\Delta}\right)^{-(\Delta-1)} = 1/\Delta, \qquad (4)$$

where we used that $d \leq \Delta - 1$ (since u_0 has a neighbour in $V \setminus U$) and that $|\lambda| \leq \lambda_\Delta$. This shows (ii). Then, we also see that $R_{H,u_0}(\lambda) \neq -1$ and so $Z_H(\lambda) \neq 0$, showing (i). This completes the induction.

To conclude the proof of the theorem we apply the same trick once more to R_{G,v_0}. From (4) we then obtain the bound $|R_{G,v_0}| < 1/(\Delta - 1)$ since v_0 may have $d = \Delta$ neighbours rather than $d \leq \Delta - 1$. Again we have $R_{G,v_0}(\lambda) \neq 1$ and so $Z_G(\lambda) \neq 0$, as desired. □

The proof essentially consists of two steps. First express a suitably chosen ratio in terms of ratios of smaller graphs. Secondly, use this expression to inductively show that these ratios are 'trapped' in some suitable region of the complex plane (the open disk of radius $1/\Delta$ in the proof above). Of course the real ingenuity comes in finding the right 'trapping region'.

This approach can be traced back to work of Dobrushin [38] and possibly even earlier. Recent years have seen many variations and refinements of this approach resulting in significant extensions of Theorem 4.1 [74, 19, 21] and zero-free regions for permanents [7, 8, 11], for the graph homomorphism partition functions [17, 16], for the partition function of the Ising and Potts models [67, 69, 75, 13, 22, 35], for Holant problems [76] and for various other graph polynomials [5, 9, 15, 14, 12, 64].

4.2 Stability of multivariate polynomials

In this subsection we briefly mention the technique of polynomial stability without going into too much detail. The basic idea here is that there are certain operations on polynomials that preserve certain useful properties. If one can use these operations to construct some desired graph polynomial or partition function from "elementary" polynomials, we can establish useful properties of the graph polynomial / partition function. The method is often most effective for multivariate polynomials, and indeed many graph polynomials have multivariate counterparts.

For our running example, the independence polynomial, the multivariate counterpart is defined as follows. Let $G = (V, E)$ be a graph and associate to each vertex v a variable x_v. The *multivariate independence polynomial* is then defined as

$$Z_G((x_v)) = \sum_{\substack{S \subseteq V \\ \text{independent}}} x^S,$$

where we use the shorthand notation $x^S := \prod_{v \in S} x_v$. Note that if we set all the variables equal to λ then we recover the original (univariate) independence polynomial. The multivariate independence polynomial is a multi-affine polynomial meaning that it is affine in each variable (i.e. if we fix all but one variable x_v it becomes a polynomial of degree 1 in x_v). It is easy to see that any multi-affine polynomial f (in the same variables $(x_v)_{v \in V}$) can be written as $f = \sum_{S \subseteq V} a_s x^S$ for some constants a_S.

For two multi-affine polynomials $P = \sum_{S \subseteq V} p_S x^S$ and $Q = \sum_{S \subseteq V} q_S x^S$, their *Schur product*, $P * Q$ is defined as the multi-affine polynomial in which the coefficient of x^S is $p_S \cdot q_S$ i.e. $P * Q = \sum_{S \subseteq V} p_S q_S x^S$. We can build up the polynomial Z_G using Schur product of simpler polynomials as follows. Suppose H_1 and H_2 are graphs on the same vertex set V and G is the union[12] of H_1 and H_2 (i.e. the edges of G are precisely the edges of H_1 together with the edges of H_2). Then

$$Z_G = Z_{H_1} * Z_{H_2}.$$

This is easy to see since we know S is an independent set of G if and only if S is an independent set of both H_1 and H_2 and the Schur product has the corresponding property that the coefficient of x^S is 1 in $Z_{H_1} * Z_{H_2}$ if and only if it is 1 in both Z_{H_1} and in Z_{H_2}. For example the 4-cycle C_4 with vertex set $\{1, 2, 3, 4\}$ and edges $\{1, 2\}$, $\{2, 3\}$, $\{3, 4\}$ and $\{4, 1\}$ is the union of two matchings M_1 with edges $\{1, 2\}, \{3, 4\}$ and M_2 with edges $\{1, 3\}, \{2, 4\}$. Using the multiplicative property[13] of the independence polynomial, we know

$$Z_{M_1} = (1 + x_1 + x_2)(1 + x_3 + x_4) \quad \text{and} \quad Z_{M_2} = (1 + x_2 + x_3)(1 + x_1 + x_4)$$

[12] This is very different from the disjoint union of graphs that we made heavy use of in Section 3.

[13] We showed this property for the univariate independence polynomial and it follows in the same way for the multivariate version

and using the Schur product property, one can check

$$Z_{C_4} = Z_{M_1} * Z_{M_2} = (1 + x_1 + x_2 + x_3 + x_4 + x_1x_3 + x_2x_4).$$

The Schur product corresponds beautifully well to taking unions of graphs for the independence polynomial, but does it preserve any useful properties? Writing $\mathbb{D}$ for the open unit disk in $\mathbb{C}$, we say a multi-affine polynomial $P = \sum_{S \subseteq V} p_S x^S$ is $\mathbb{D}$-stable if $P((x_v)_{v \in V}) \neq 0$ whenever $x_v \in \mathbb{D}$ for all $v \in V$. It is well known (see [6]) that if P and Q are $\mathbb{D}$-stable then so is $P * Q$. This seems promising for us, but unfortunately, the independence polynomial of a matching or indeed a single edge (out of which we build all other independence polynomials) is not $\mathbb{D}$-stable, e.g. $Z_{M_1}(-\frac{1}{2}, -\frac{1}{2}, 0, 0) = 0$. The independence polynomial of a matching is however non-zero if all the arguments are in an open disk of radius $1/2$. Now, using the fact that every graph in $\mathcal{G}_\Delta$ is the union of at most $\Delta + 1$ matchings (Vizing's theorem) and applying a simple scaling argument, one can still make use of the $\mathbb{D}$-stability of Schur products to show that Z_G is non-zero if all arguments are in a disk of radius smaller than $1/2^{\Delta+1}$, where $G \in \mathcal{G}_\Delta$.

This is a much weaker bound than Theorem 4.1 from the the previous subsection, but is given simply to illustrate the idea of stability. The idea of using multi-affine polynomials and operations preserving zero-freeness was pioneered by Asano [2] about fifty years ago to give a short and elegant proof of the famous Lee-Yang theorem (see also [6] for a proof using Schur products.) The theorem states that the partition function of the Ising model (in terms of vertex activities), which essentially is the generating function of the edge cuts in the graph, has all its zeros on the unit circle under suitable conditions; we choose not to introduce the relevant background here. By now there are several variations of the technique, some of which use the Grace-Szëgo-Walsh theorem, and they have been applied to partition functions of several models and graph polynomials [80, 80, 88, 54, 54, 20].

4.3 The polymer method

We introduced the multivariate independence polynomial in the last subsection to illustrate the idea of polynomial stability. It turns out that many other graph polynomials and partition functions can be expressed as evaluations of multivariate independence polynomials of a particular type. For this reason, there has been a lot of interest in understanding and proving conditions that guarantee zero-freeness of such multivariate independence polynomials. This idea of first rewriting a partition function/graph polynomial as an evaluation of a multivariate independence polynomial and then checking conditions from the literature known to guarantee that the latter evaluation is nonzero is a powerful technique originating in statistical physics. There, the multivariate independence polynomial is sometimes called the partition function of a polymer model, and the technique we describe is sometimes called the polymer method.

We will give an example of this idea applied to the chromatic polynomial, a graph polynomial used for counting proper colourings of a graph, which we will shortly introduce. We sketch a proof of a result of Férnandez and Procacci [44] and Jackson, Procacci and Sokal [59] about zero-freeness of the chromatic polynomial. At the end of the subsection, we list some recent results based on this technique and indicate how a variation of this technique can in fact be used directly to design efficient algorithms to approximate graph polynomials, without having to use the interpolation method.

4.3.1 The chromatic polynomial

For a graph $G = (V, E)$ and integer q, a proper q colouring of G is an assignment of q colours (usually labelled $1, \dots, q$) to the vertices such that adjacent vertices receive different colours. This means in particular that all vertices assigned some fixed colour i form an independent set. The function χ_G counts the number of proper q-colourings of G, that is, for each $q \in \mathbb{N}$, $\chi_G(q)$ is defined to be the number of proper q-colourings of G. For example the number of proper q-colourings of a triangle is $q(q-1)(q-2)$ since after ordering the vertices arbitrarily, the first vertex can receive any of the q colours, the second vertex may receive any of the colours except the colour of the first vertex, and the third vertex may receive any colour except those of the first two vertices (which are different).[14] More generally, the number of proper q colourings of K_r, the complete graph on r vertices is $q(q-1)\cdots(q-r+1)$, i.e. $\chi_{K_r}(q) = q(q-1)\cdots(q-r+1)$ for every $q \in \mathbb{N}$. For any tree T on r vertices, $\chi_T(q) = q(q-1)^{r-1}$ for all $q \in \mathbb{N}$ since if we colour the vertices in a breadth-first ordering, then the first vertex may receive any of the q colours, while each subsequent vertex can receive any colour except that of its parent. Of course, it is not usually so easy to determine $\chi_G(q)$ because it is NP-complete to decide if there is even one proper q-colouring of G, i.e. whether $\chi_G(q)$ is positive or not. Nonetheless, as the examples above suggest, $\chi_G(q)$ is always a polynomial in q as we shall see shortly, and χ_G is called the chromatic polynomial of G.

The chromatic polynomial was introduced in 1912 by Birkhoff in an attempt to prove the four colour theorem. It has a long history and has been studied from many perspectives together with its far-reaching generalisation, the Tutte polynomial (see [40, 43] for a comprehensive account).

We now establish a very useful formula for the chromatic polynomial called the random cluster model, due to Fortuin and Kasteleyn (see [45]); it is sometimes used as the definition of the chromatic polynomial. Formally, a proper q-colouring of a graph $G = (V, E)$ is a function $f : V \to \{1, \dots, q\} =: [q]$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$. Then we can write

$$\chi_G(q) = \sum_{f: V \to [q]} \prod_{\{u,v\} \in E} \mathbb{1}_{f(u) \neq f(v)},$$

where $\mathbb{1}_{f(u) \neq f(v)}$ is the indicator function that $f(u) \neq f(v)$ (so that the product is 1 if and only if all edges are properly coloured). Replacing $\mathbb{1}_{f(u) \neq f(v)}$ with $(1 - \mathbb{1}_{f(u) = f(v)}))$

[14] Note that the formula is correct even when $q < 3$, i.e. when there are no proper q-colourings of the triangle.

and expanding, we obtain

$$\begin{aligned}\chi_G(q) = \sum_{f:V\to[q]} \prod_{\{u,v\}\in E} (1 - \mathbb{1}_{f(u)=f(v)})) &= \sum_{f:V\to[q]} \sum_{F\subseteq E} (-1)^{|F|} \prod_{\{u,v\}\in F} \mathbb{1}_{f(u)=f(v)} \\ &= \sum_{F\subseteq E} (-1)^{|F|} \sum_{f:V\to[q]} \prod_{\{u,v\}\in F} \mathbb{1}_{f(u)=f(v)}.\end{aligned}$$

The inner sum in the last expression is equal to $q^{k(F)}$, where $k(F)$ is the number of components of the graph (V,F). The reason is that the product is 1 for an assignment f if and only if every edge of F is monochromatic in f, which means that f must assign a single colour to each component of (V,F). There are precisely $q^{k(F)}$ ways of doing this. Thus

$$\chi_G(q) := \sum_{F\subseteq E} q^{k(F)}(-1)^{|F|}, \tag{5}$$

and so we see that χ_G is indeed a polynomial (although there are easier ways of showing this) and has degree $|G|$.

Our goal will be to prove the following zero-freeness result for the chromatic polynomial.

Theorem 4.2 ([44, 59]). *Let G be any graph. Then all the zeros of χ_G are contained in the disk of radius $6.91\Delta(G)$ centered at 0 in the complex plane.*

It is likely that the constant 6.91 can be improved, but it is not clear what the optimal value is likely to be; see [79, Footnote 4] for further discussion. By the Taylor polynomial interpolation method, Theorem 4.2 almost immediately implies an FPTAS for approximating $\chi_G(q)$ whenever $G \in \mathcal{G}_\Delta$ and $|q| \geq 6.92\Delta$. The trick is to apply the interpolation method to the polynomial $q^{|V|}\chi_G(1/q)$, which has no zeros in the disk of radius $\frac{1}{6.91\Delta}$. From the combinatorial perspective, this implies an FPTAS to count the number of proper q-colourings of any graph $G \in \mathcal{G}_\Delta$ whenever $q > 6.91\Delta$. It is believed that there is an FPTAS for counting proper q-colourings whenever $q > \Delta$ and this is an active area of research. By proving a zero-freeness result for a different polynomial (the partition function of the Potts model) Liu, Sinclair, and Srivastava [66] have shown that there is an FPTAS when $q \geq 2\Delta$, and this is currently the state of the art.[15]

We now sketch the proof of Theorem 4.2. As mentioned, we will need to work again with the (multivariate) independence polynomial and to make use of a suitable zero-freeness result for it.

4.3.2 The chromatic polynomial as a multivariate independence polynomial

Our first lemma shows how to express the chromatic polynomial of a graph G as an evaluation of the multivariate independence polynomial of an associated graph. For this we need some notation. Let $G = (V,E)$ be a graph. Define a new graph Γ whose

[15] There are improved bounds if we allow randomised algorithms based on the Markov chain Monte Carlo method [87, 33].

vertices are subsets S of V of size at least two. (In the context of the polymer method, these sets are called polymers.) Two of those sets S, T are connected by an edge if and only if $S \cap T \neq \emptyset$. Notice that the graph Γ is independent of the edges of G.

We now associate weights to vertices of Γ as follows; these will depend on the edges of G and on q (the variable in the chromatic polynomial). For each vertex S of Γ, i.e. $S \subseteq V$ with $|S| \geq 2$, define

$$\lambda_S := \sum_{\substack{F \subseteq E(S) \\ \text{connected}}} (-1)^{|F|} q^{|S|-1}. \tag{6}$$

Now the multivariate independence polynomial of Γ with the (complex) vertex weights λ_S is given by

$$Z_\Gamma((\lambda_S)) = \sum_{\substack{I \subseteq V(\Gamma) \\ \text{independent}}} \prod_{S \in I} \lambda_S. \tag{7}$$

Lemma 4.3. *With notation as above we have*

$$q^{|V|}\chi_G(1/q) = Z_\Gamma((\lambda_S)).$$

Proof. We start by expanding the left-hand side using (5)

$$q^{|V|}\chi_G(1/q) = \sum_{F \subseteq E} (-1)^{|F|} q^{|V|-k(F)} = \sum_{F \subseteq E} \prod_{C \text{ component } of F} (-1)^{|C|} q^{|V(C)|-1}.$$

Next, we break up the sum over $F \subseteq E$ in terms of the component structure of F as follows. We sum over all F that have exactly k connected components with vertex sets $S_1, \ldots, S_k$ and then we sum over all possible choices of $S_1, \ldots, S_k$ and all possible choices of k. In fact we can ignore the components that consist of a single vertex (and no edge) since they contribute a factor of 1 to the product above. In this way (after exchanging a sum and product) we obtain

$$q^{|V|}\chi_G(1/q) = \sum_{k \geq 0} \sum_{\substack{S_1, \ldots, S_k \subseteq V \\ S_i \cap S_j = \emptyset \text{ for } i \neq j \\ |S_i| \geq 2}} \prod_{i=1}^{k} \sum_{\substack{F_i \subseteq E(S_i) \\ (S_i, F_i) \text{ connected}}} (-1)^{|F_i|} q^{|S_i|-1}.$$

By construction any collection of sets $\{S_1, \ldots, S_k\}$ contributing to this sum forms an independent set of size k in the graph Γ. The weights are constructed precisely so that the last expression is $Z_\Gamma((\lambda_S))$, as desired. □

4.3.3 Zero-freeness conditions and their verification

Here we present a result due to Biascot, Férnandez and Procacci [25] that provides useful conditions that guarantee that our multivariate independence polynomial (for graphs of the type Γ) does not evaluate to zero. We will then verify these conditions for our situation. Let $G = (V, E)$ and Γ be as before.

Theorem 4.4 ([25]). *For any complex numbers $(\lambda_S)_{S\in V(\Gamma)}$ and any $a > 1$, if, for each $v \in V$, it holds that*

$$\sum_{\substack{S|v\in S\\ |S|\geq 2}} |\lambda_S| a^{|S|} \leq a - 1, \tag{8}$$

then $Z_\Gamma((\lambda_S)) \neq 0$.

The theorem can be proved along the same lines as the proof of Theorem 4.1. See [25, Proposition 3.1] for a proof along these lines and a discussion of how this condition compares with other similar conditions including the Kotécky-Preis conditions [63] and Dobrushin's conditions [38].

To verify the conditions in Theorem 4.4, we need a bound on the weights λ_S given in (6). Our first step in this direction is to get rid of the 'alternating signs' in (6). The lemma below can for example be proved using well-known properties of the Tutte polynomial; see e.g. [43] for these properties and see [73, 85] for a direct proof.

Lemma 4.5. *Let H be a connected graph and denote by $\tau(H)$ the number of spanning trees in H. Then*

$$\Bigg| \sum_{\substack{F\subseteq E(H)\\ (V(H),F) \text{ connected}}} (-1)^{|F|} \Bigg| \leq \tau(H).$$

For a graph $G = (V, E)$, a vertex $v \in V$, and a variable x we define the *tree generating function* by

$$T_{G,v}(x) := \sum_{\substack{T\subseteq E(G)\\ (V(T),T) \text{ is a tree, } v\in V(T)}} x^{|T|}.$$

We can now bound $\sum_{S|v\in S,|S|\geq 2} |\lambda_S| a^{|S|}$ in terms of the tree generating function as follows:

$$\begin{aligned}
\sum_{S|v\in S,|S|\geq 2} |\lambda_S| a^{|S|} &= \sum_{S|v\in S,|S|\geq 2} \Bigg| \sum_{\substack{F\subseteq E(S)\\ \text{connected}}} (-1)^{|F|} q^{|S|-1} \Bigg| a^{|S|} \\
&\leq \sum_{S|v\in S,|S|\geq 2} \Bigg| \sum_{\substack{F\subseteq E(S)\\ \text{connected}}} (-1)^{|F|} \Bigg| |q|^{|S|-1} a^{|S|} \\
&\leq \sum_{S|v\in S,|S|\geq 2} \tau(G[S]) |q|^{|S|-1} a^{|S|} \\
&= a T_{G,v}(a|q|) - a.
\end{aligned} \tag{9}$$

The next lemma shows how to bound the tree generating function. The proof we give is slightly shorter than the proof given in [59], and is new as far as we know.

Lemma 4.6 ([58]). *Let $G = (V, E)$ be a graph of maximum degree at most $\Delta \geq 1$ and let $v \in V$. Fix any $\alpha > 1$. Then*

$$T_{G,v}\left(\frac{\ln \alpha}{\alpha\Delta}\right) \leq \alpha.$$

Proof. The proof is by induction on the number of vertices of G. If $|V| = 1$, the statement is clearly true. Next assume that $|V| \geq 2$. Given a tree T such that $v \in V(T)$ let S be the set of neighbours of v in $V(T)$. After removing v from T, the tree decomposes into the disjoint union of $|S|$ trees, each containing a unique vertex from S. Therefore, writing $c = \frac{\ln \alpha}{\alpha \Delta}$, we have

$$T_{G,v}(c) \leq \sum_{S \subseteq N_G(v)} c^{|S|} \prod_{s \in S} T_{G-v,s}(c),$$

which by induction is bounded by

$$\sum_{S \subseteq N_G(V)} (c\alpha)^{|S|} \leq (1 + (\ln \alpha)/\Delta)^{\Delta} \leq e^{\ln \alpha} = \alpha.$$

This finishes the proof. □

We now combine all our ingredients to finish the proof of Theorem 4.2. Fix $\Delta \geq 2$. For $a > 1$ to be determined, define $\alpha = \alpha(a) = 2 - 1/a$. Then if

$$|q| \leq \frac{\ln \alpha}{a\alpha\Delta} = \frac{\ln(2 - 1/a)}{(2a - 1)\Delta},$$

we have $\chi_G(1/q) \neq 0$ for any graph of maximum degree at most Δ. Indeed, for such a value of $q \neq 0$ we have $|aq| \leq \frac{\ln \alpha}{\alpha\Delta}$ and therefore by (9) and the previous lemma

$$\sum_{S | v \in S, |S| \geq 2} |\lambda_S| a^{|S|} \leq a(T_{G,v}(a|q|) - 1) \leq a(\alpha - 1) = a(1 - 1/a) = a - 1,$$

and so by Theorem 4.4 and Lemma 4.3 we have $\chi_G(1/q) \neq 0$. In other words if $|q| \geq \Delta \frac{2a-1}{\ln(2-1/a)}$ we have $\chi_G(q) \neq 0$. One can determine

$$\min_{a>1} \frac{2a - 1}{\ln(2 - 1/a)} < 6.91,$$

where the minimum is attained at $a \cong 1.588$. This finishes the proof sketch of Theorem 4.2.

4.3.4 Recipe and relation to cluster expansion

The steps we took to prove Theroem 4.2 suggest a 'recipe' for proving absence of zeros using the polymer approach:

- Express the graph polynomial as an evaluation of the multivariate independence polynomial of an associated graph.
- Use the conditions from Theorem 4.4 (or other conditions) that guarantee the evaluation is nonzero.

- Verify these conditions using combinatorial arguments.

Most combinatorial applications of this 'recipe' include various extensions and variations of the chromatic polynomial. See [85, 44, 39, 60, 59, 41, 36, 35] for some examples in this direction.

From a statistical physics perspective both Theorem 4.2 and Theorem 4.1 are statements about so-called high temperature models (in the case of Theorem 4.1, high temperature means small values of λ for the independence polynomial). Surprisingly, for some restricted families of graphs, the 'recipe' above can also sometimes be used at low temperature (see e.g. [28, 46] for this in statistical physics). For example, it has been used in combination with the interpolation method to design efficient approximation algorithms to approximate the independence polynomial at large λ on certain subgraphs of the integer lattice $\mathbb{Z}^d$ [57], but also on bipartite expander graphs [61]. In fact, [61] slightly modified the approach from [57]. The idea is to use conditions like those in Theorem 4.4 to show absolute convergence of the *cluster expansion*, a formal power series of the logarithm of $Z_\Gamma((\lambda_S))$, and to bound the remainder after truncating it at a suitable depth. This avoids the use of the interpolation method and may occasionally lead to faster algorithms, but other than that is quite similar in spirit. See [27, 30, 65, 55, 31, 71, 50, 47, 56, 32] for some results inspired by and based on this.

5 Concluding remarks

We have shown how absence of zeros allows one to design efficient algorithms to approximately compute evaluations of graph polynomials using Barvinok's interpolation method. A key part of this method is establishing absence of zeros for the graph polynomials in question. A few natural questions that remain are: how do other approaches for approximate counting relate to absence of zeros, and what does presence of zeros mean for the possibility to design efficient approximation algorithms. In this section we will briefly address these two questions pointing the interested reader to the relevant literature.

5.1 Absence of zeros and other algorithm approaches

As mentioned in the introduction there are two other (and older) approaches for designing approximation algorithms to compute evaluations of graph polynomials: a Markov chain based sampling approach and the method of correlation decay. We will not discuss the workings of these approaches here, but we mention how these approaches relate to the interpolation method, or rather, how they relate to absence of zeros.

Recently it was shown that a standard technique for proving decay of correlations can be transformed to prove absence of zeros near the real axis [82, 66]. In the other direction, some results appeared indicating that absence of zeros can be used to establish some form of decay of correlations [52, 77]. Perhaps more surprisingly, in [1, 34] it was shown that if a multivariate version of the polynomial has no zeros near the positive real axis, then the associated Glauber dynamics (a local Markov chain often used in approximate counting and sampling) mixes rapidly. These results indicate that, while absence of complex zeros is vital for the interpolation method, it also plays a key role (albeit in disguise) in these two other approaches for approximate counting.

5.2 Presence of zeros

In this section we discuss how presence of zeros is related to hardness of approximation. We will again specialise the discussion to the independence polynomial and give some references to results on other polynomials at the end of this section. In what follows we shall see that presence of zeros implies hardness of approximating the independence polynomial.

Let us first state the precise algorithmic problem in question. Let $\lambda \in \mathbb{Q}[i]$ (the set of complex numbers whose real and imaginary parts are both rational) and let $\Delta \in \mathbb{N}$. Consider the following computational problem.

Name #Hard-CoreNorm(λ, Δ)

Input A graph G of maximum degree at most Δ.

Output If $Z_G(\lambda) \neq 0$ the algorithm must output a rational number N such that $N/1.001 \leq |Z_G(\lambda)| \leq 1.001N$. If $Z_G(\lambda) = 0$ the algorithm may output any rational number.

It is easy to show that replacing the constant 1.001 by any other constant $C > 1$ does not change the complexity of the problem.[16]

The typical notion of hardness one considers for computational counting problems is #P-completeness/hardness. We do not introduce the notion formally, but wish to impress only that one does not expect a polynomial-time algorithm for a #P-complete counting problem (just as one does not expect a polynomial-time algorithm for an NP-complete problem). For example, the problem of exactly counting the number of independent sets of a graph of maximum degree Δ is known to be #P-complete [78, 86, 42] for any $\Delta \geq 3$.

Returning to the problem #Hard-CoreNorm(λ, Δ), we define the sets

$$\begin{aligned} \mathcal{P}_\Delta &= \{\lambda \in \mathbb{Q}[i] \mid \text{ \#Hard-CoreNorm}(\lambda, \Delta) \text{ is \#P-hard}\}, \\ \mathcal{Z}_\Delta &= \{\lambda \in \mathbb{C} \mid Z_G(\lambda) = 0 \text{ for some graph } G \in \mathcal{G}_\Delta\}. \end{aligned}$$

Building on [23], it was shown in [37] that the closure of the set $\mathcal{Z}_\Delta$ is contained in the closure of the set $\mathcal{P}_\Delta$, meaning that arbitrarily close to any zero $\lambda \in \mathcal{Z}_\Delta$ there exists

[16] An algorithm that solves the problem above in polynomial time can also be used to solve the problem with 1.001 replaced by $(1.001)^2$ by running the original algorithm on the disjoint union of two copies of the graph.

a parameter $\lambda' \in \mathbb{Q}[i]$ such that #Hard-CoreNorm(λ', Δ) is #P-hard. (A similar result holds if instead of approximating the norm one wishes to approximate the argument of $Z_G(\lambda)$.) Recall that Theorem 4.1 gives a zero-free region for the independence polynomial that contains the point 0. If one can show that the *maximal* zero-free region of the independence polynomial for bounded degree graphs is connected, then this would result in an essentially complete understanding of the hardness of approximating the independence polynomial at complex parameters on bounded degree graphs [37]. We remark that quite recently it was shown that in the $\Delta \to \infty$ limit this is true [18], but unfortunately this does not give us information for any finite Δ yet.

For some models/polynomials such as the matching polynomial [24] and the ferromagnetic Ising model (as a function of the external field) [29] we know that absence/presence of zeros on bounded degree graphs exactly corresponds to easiness/ hardness of approximation. For others such as the Ising model (as a function of edge interaction) [48] a partial correspondence has been established. This suggests a program of study to understand this connection for more models and also to understand the phenomenon in general.

The interpolation method is clearly a powerful technique for establishing fast approximation algorithms to evaluate graph polynomials at complex parameters, but more than that, it often seems to capture the dichotomy between parameters where approximate computation is easy and hard.

References

[1] Yeganeh Alimohammadi, Nima Anari, Kirankumar Shiragur, and Thuy-Duong Vuong. Fractionally log-concave and sector-stable polynomials: counting planar matchings and more. *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 433–446.

[2] Taro Asano. Lee-Yang theorem and the Griffiths inequality for the anisotropic Heisenberg ferromagnet. *Phys. Rev. Lett.*, 24:1409–1411, 1970.

[3] Zonglei Bai, Yongzhi Cao, and Hanpin Wang. Zero-freeness and approximation of real Boolean Holant problems. *Theoretical Computer Science*, 917:12–30, 2022.

[4] Antar Bandyopadhyay and David Gamarnik. Counting without sampling: asymptotics of the log-partition function for certain statistical physics models. *Random Structures & Algorithms*, 33(4):452–479, 2008.

[5] Alexander Barvinok. Computing the partition function for cliques in a graph. *Theory Comput.*, 11:339–355, 2015.

[6] Alexander Barvinok. *Combinatorics and Complexity of Partition Functions*, volume 30 of *Algorithms and combinatorics*. Springer, 2016.

[7] Alexander Barvinok. Computing the permanent of (some) complex matrices. *Found. Comput. Math.*, 16(2):329–342, 2016.

[8] Alexander Barvinok. Approximating permanents and hafnians. *Discrete Anal.*, Paper No. 2, 34, 2017.

[9] Alexander Barvinok. Computing the partition function of a polynomial on the Boolean cube. In *A journey through discrete mathematics*, pages 135–164, 2017.

[10] Alexander Barvinok. Approximating real-rooted and stable polynomials, with combinatorial applications. *Online J. Anal. Comb.*, (14):Paper No. 8, 13, 2019.

[11] Alexander Barvinok. Computing permanents of complex diagonally dominant matrices and tensors. *Israel J. Math.*, 232(2):931–945, 2019.

[12] Alexander Barvinok. Stability and complexity of mixed discriminants. *Math. Comp.*, 89(322):717–735, 2020.

[13] Alexander Barvinok and Nicholas Barvinok. More on zeros and approximation of the Ising partition function. *Forum Math. Sigma*, 9: Paper No. e46, 18, 2021.

[14] Alexander Barvinok and Anthony Della Pella. Testing for dense subsets in a graph via the partition function. *SIAM J. Discrete Math.*, 34(1):308–327, 2020.

[15] Alexander Barvinok and Guus Regts. Weighted counting of solutions to sparse systems of equations. *Combin. Probab. Comput.*, 28(5):696–719, 2019.

[16] Alexander Barvinok and Pablo Soberón. Computing the partition function for graph homomorphisms with multiplicities. *J. Combin. Theory Ser. A*, 137:1–26, 2016.

[17] Alexander Barvinok and Pablo Soberón. Computing the partition function for graph homomorphisms. *Combinatorica*, 37(4):633–650, 2017.

[18] Ferenc Bencs, Pjotr Buys, and Han Peters. The limit of the zero locus of the independence polynomial for bounded degree graphs. *arXiv preprint arXiv:2111.06451*, 2021.

[19] Ferenc Bencs and Péter Csikvári. Note on the zero-free region of the hard-core model. *arXiv preprint arXiv:1807.08963*, 2018.

[20] Ferenc Bencs, Péter Csikvári, and Guus Regts. Some applications of Wagner's weighted subgraph counting polynomial. *Electron. J. Combin.*, 28(4):Paper No. 4.14, 21, 2021.

[21] Ferenc Bencs, Péter Csikvári, Piyush Srivastava, and Jan Vondrák. On complex roots of the independence polynomial. *arXiv preprint arXiv:2204.04868*, 2022.

[22] Ferenc Bencs, Ewan Davies, Viresh Patel, and Guus Regts. On zero-free regions for the anti-ferromagnetic Potts model on bounded-degree graphs. *Ann. Inst. Henri Poincaré D*, 8(3):459–489, 2021.

[23] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, and Daniel Štefankovič. Inapproximability of the independent set polynomial in the complex plane. *SIAM J. Comput.*, 49(5):STOC18–395–STOC18–448, 2020.

[24] Ivona Bezáková, Andreas Galanis, Leslie Ann Goldberg, and Daniel Štefankovič. The complexity of approximating the matching polynomial in the complex plane. *ACM Trans. Comput. Theory*, 13(2):Art. 13, 37, 2021.

[25] Rodrigo Bissacot, Roberto Fernández, and Aldo Procacci. On the convergence of cluster expansions for polymer gases. *J. Stat. Phys.*, 139(4):598–617, 2010.

[26] Magnus Bordewich, Michael Freedman, Lázalo Lovász, and Dominic Welsh. Approximate counting and quantum computation. *Combin. Probab. Comput.*, 14(5-6):737–754, 2005.

[27] Christian Borgs, Jennifer Chayes, Tyler Helmuth, Will Perkins, and Prasad Tetali. Efficient sampling and counting algorithms for the Potts model on $\mathbb{Z}^d$ at all temperatures. *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 738–751, 2020.

[28] Christian Borgs and John Imbrie. A unified approach to phase diagrams in field theory and statistical mechanics. *Comm. Math. Phys.*, 123(2):305–328, 1989.

[29] Pjotr Buys, Andreas Galanis, Viresh Patel, and Guus Regts. Lee-Yang zeros and the complexity of the ferromagnetic Ising model on bounded-degree graphs. *Forum Math. Sigma*, 10:Paper No. e7, 43, 2022.

[30] Sarah Cannon and Will Perkins. Counting independent sets in unbalanced bipartite graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms*, pages 1456–1466, 2020.

[31] Charles Carlson, Ewan Davies, and Alexandra Kolla. Efficient algorithms for the Potts model on small-set expanders. *arXiv preprint arXiv:2003.01154*, 2020.

[32] Katrin Casel, Philipp Fischbeck, Tobias Friedrich, Andreas Göbel, and J. A. Gregor Lagodzinski. Zeros and approximations of Holant polynomials on the complex plane. *Comput. Complexity*, 31(2):Paper No. 11, 2022.

[33] Sitan Chen, Michelle Delcourt, Ankur Moitra, Guillem Perarnau, and Luke Postle. Improved bounds for randomly sampling colorings via linear programming. *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2216–2234, 2019.

[34] Zongchen Chen, Kuikui Liu, and Eric Vigoda. Spectral independence via stability and applications to Holant-type problems. *IEEE 62nd Annual Symposium on Foundations of Computer Science*, pages 149–160, 2022.

[35] Matthew Coulson, Ewan Davies, Alexandra Kolla, Viresh Patel, and Guus Regts. Statistical physics approaches to unique games. *35th Computational Complexity Conference*, volume 169 of *LIPIcs. Leibniz Int. Proc. Inform.*, Art. No. 13, 27. 2020.

[36] Péter Csikvári and Péter Frenkel. Benjamini-Schramm continuity of root moments of graph polynomials. *European J. Combin.*, 52(part B):302–320, 2016.

[37] David de Boer, Pjotr Buys, Lorenzo Guerini, Han Peters, and Guus Regts. Zeros, chaotic ratios and the computational complexity of approximating the independence polynomial. *arXiv preprint arXiv:2104.11615*, 2021.

[38] R. L. Dobrushin. Estimates of semi-invariants for the Ising model at low temperatures. In *Topics in statistical and theoretical physics*, volume 177 of *Amer. Math. Soc. Transl. Ser. 2*, pages 59–81, 1996.

[39] F. M. Dong and K. M. Koh. Bounds for the real zeros of chromatic polynomials. *Combin. Probab. Comput.*, 17(6):749–759, 2008.

[40] F. M. Dong, K. M. Koh, and K. L. Teo. *Chromatic polynomials and chromaticity of graphs*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005.

[41] Fengming Dong and Xian'an Jin. Zeros of Jones polynomials of graphs. *Electron. J. Combin.*, 22(3):Paper 3.23, 18, 2015.

[42] Martin Dyer and Catherine Greenhill. On Markov chains for independent sets. *J. Algorithms*, 35(1):17–49, 2000.

[43] Joanna A. Ellis-Monaghan and Iain Moffatt. *Handbook of the Tutte Polynomial and Related Topics*. CRC Press, 2022.

[44] Roberto Fernández and Aldo Procacci. Regions without complex zeros for chromatic polynomials on graphs with bounded degree. *Combin. Probab. Comput.*, 17(2):225–238, 2008.

[45] C. M. Fortuin and P. W. Kasteleyn. On the random-cluster model. I. Introduction and relation to other models. *Physica*, 57:536–564, 1972.

[46] S. Friedli and Y. Velenik. *Statistical mechanics of lattice systems: a concrete mathematical introduction.* Cambridge University Press, Cambridge, 2018.

[47] Tobias Friedrich, Andreas Göbel, Martin S Krejca, and Marcus Pappik. Polymer dynamics via cliques: New conditions for approximations. *arXiv preprint arXiv:2007.08293*, 2020.

[48] Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued Ising model on bounded degree graphs. *arXiv preprint arXiv:2105.00287*, 2021.

[49] Andreas Galanis, Leslie Ann Goldberg, and Andrés Herrera-Poyatos. The complexity of approximating the complex-valued Potts model. *Comput. Complexity*, 31(1):Paper No. 2, 94, 2022.

[50] Andreas Galanis, Leslie Ann Goldberg, and James Stewart. Fast algorithms for general spin systems on bipartite expanders. *ACM Trans. Comput. Theory*, 13(4):Art. 25, 18, 2021.

[51] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models. *Combin. Probab. Comput.*, 25(4):500–559, 2016.

[52] David Gamarnik. Correlation decay and the absence of zeros property of partition functions. *Random Structures & Algorithms*, 2022.

[53] David Gaunt and Michael Fisher. Hard-sphere lattice gases. i. plane-square lattice. *J. Chem. Phys.*, 43(8):2840–2863, 1965.

[54] Heng Guo, Chao Liao, Pinyan Lu, and Chihao Zhang. Zeros of Holant problems: locations and algorithms. *ACM Trans. Algorithms*, 17(1):Art. 4, 25, 2021.

[55] Tyler Helmuth, Matthew Jenssen, and Will Perkins. Finite-size scaling, phase coexistence, and algorithms for the random cluster model on random graphs. *arXiv preprint arXiv:2006.11580*, 2020.

[56] Tyler Helmuth and Ryan L Mann. Efficient algorithms for approximating quantum partition functions at low temperature. *arXiv preprint arXiv:2201.06533*, 2022.

[57] Tyler Helmuth, Will Perkins, and Guus Regts. Algorithmic Pirogov-Sinai theory. *Probab. Theory Related Fields*, 176(3-4):851–895, 2020.

[58] Jeroen Huijben and Guus Regts. Private communication. 2021.

[59] Bill Jackson, Aldo Procacci, and Alan D. Sokal. Complex zero-free regions at large $|q|$ for multivariate Tutte polynomials (alias Potts-model partition functions) with general complex edge weights. *J. Combin. Theory Ser. B*, 103(1):21–45, 2013.

[60] Bill Jackson and Alan D. Sokal. Zero-free regions for multivariate Tutte polynomials (alias Potts-model partition functions) of graphs and matroids. *J. Combin. Theory Ser. B*, 99(6):869–903, 2009.

[61] Matthew Jenssen, Peter Keevash, and Will Perkins. Algorithms for #BIS-hard problems on expander graphs. *SIAM J. Comput.*, 49(4):681–710, 2020.

[62] Mark Jerrum. *Counting, sampling and integrating: algorithms and complexity*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, 2003.

[63] R. Kotecký and D. Preiss. Cluster expansion for abstract polymer models. *Comm. Math. Phys.*, 103(3):491–498, 1986.

[64] Liang Li and Guangzeng Xie. Complex contraction on trees without proof of correlation decay. *arXiv preprint arXiv:2112.15347*, 2021.

[65] Chao Liao, Jiabao Lin, Pinyan Lu, and Zhenyu Mao. An FPTAS for the hardcore model on random regular bipartite graphs. *Theoretical Computer Science*, 929:174–190, 2022.

[66] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. Correlation decay and partition function zeros: Algorithms and phase transitions. *SIAM Journal on Computing*, 0(0):FOCS19–200–FOCS19–252, 0.

[67] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. Fisher zeros and correlation decay in the Ising model. *J. Math. Phys.*, 60(10):103304, 12, 2019.

[68] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. The Ising partition function: zeros and deterministic approximation. *J. Stat. Phys.*, 174(2):287–315, 2019.

[69] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. A deterministic algorithm for counting colorings with 2Δ colors. *IEEE 60th Annual Symposium on Foundations of Computer Science*, pages 1380–1404, 2019.

[70] Ryan L. Mann and Michael J. Bremner. Approximation Algorithms for Complex-Valued Ising Models on Bounded Degree Graphs. *Quantum*, 3:162, July 2019.

[71] Ryan L. Mann and Tyler Helmuth. Efficient algorithms for approximating quantum partition functions. *J. Math. Phys.*, 62(2):Paper No. 022201, 7, 2021.

[72] Viresh Patel and Guus Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM J. Comput.*, 46(6):1893–1919, 2017.

[73] Oliver Penrose. Convergence of fugacity expansions for classical systems. *Statistical mechanics: foundations and applications*, page 101, 1967.

[74] Han Peters and Guus Regts. On a conjecture of Sokal concerning roots of the independence polynomial. *Michigan Math. J.*, 68(1):33–55, 2019.

[75] Han Peters and Guus Regts. Location of zeros for the partition function of the Ising model on bounded degree graphs. *J. Lond. Math. Soc. (2)*, 101(2):765–785, 2020.

[76] Guus Regts. Zero-free regions of partition functions with applications to algorithms and graph limits. *Combinatorica*, 38(4):987–1015, 2018.

[77] Guus Regts. Absence of zeros implies strong spatial mixing. *arXiv preprint arXiv:2111.04809*, 2021.

[78] Dan Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.

[79] Gordon F. Royle and Alan D. Sokal. Linear bound in terms of maxmaxflow for the chromatic roots of series-parallel graphs. *SIAM J. Discrete Math.*, 29(4):2117–2159, 2015.

[80] David Ruelle. Zeros of graph-counting polynomials. *Comm. Math. Phys.*, 200(1):43–56, 1999.

[81] Alexander D. Scott and Alan D. Sokal. The repulsive lattice gas, the independent-set polynomial, and the Lovász local lemma. *J. Stat. Phys.*, 118(5-6):1151–1261, 2005.

[82] Shuai Shao and Yuxin Sun. Contraction: a unified perspective of correlation decay and zero-freeness of 2-spin systems. *J. Stat. Phys.*, 185(2):Paper No. 12, 25, 2021.

[83] J. B. Shearer. On a problem of Spencer. *Combinatorica*, 5(3):241–245, 1985.

[84] Allan Sly and Nike Sun. Counting in two-spin models on d-regular graphs. *Ann. Probab.*, 42(6):2383–2416, 2014.

[85] Alan D. Sokal. Bounds on the complex zeros of (di)chromatic polynomials and Potts-model partition functions. *Combin. Probab. Comput.*, 10(1):41–77, 2001.

[86] Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2):398–427, 2001.

[87] Eric Vigoda. Improved bounds for sampling colorings. *J. Math. Phys.*, 41(3):1555–1569, 2000.

[88] David G. Wagner. Weighted enumeration of spanning subgraphs with degree constraints. *J. Combin. Theory Ser. B*, 99(2):347–357, 2009.

[89] Dror Weitz. Counting independent sets up to the tree threshold. *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 140–149, 2006.

[90] D. J. A. Welsh. *Complexity: knots, colourings and counting*, volume 186 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 1993.

The Distributed Computing Column

Seth Gilbert
National University of Singapore
seth.gilbert@comp.nus.edu.sg

Clock synchronization is one of the fundamental problems in distributed computing, playing a critical role at one of the lowest levels of the protocol stack. As such, it is a basic building block that many higher level protocols depend on (and many algorithm designers take for granted). It is a long-studied problem, with foundational results going back decades and decades. And it is a basic service that is used by numerous real-world protocols (e.g., in the form of NTP).

In this column, Swen Jacobs and Christoph Lenzen give an overview of the state of the art, as well as argue that there remains a lot of work left to be done! They focus on the problem of robustness: most current synchronization protocols are not secure, and most are susceptible to fairly simple attacks. Protocols like NTP are well-known to be easy to attack, and can have severe consequences for real-world systems like power grids and financial networks. Jacobs and Lenzen discuss a variety of open problems in the area of (robust) clock synchronization, including issues of redundancy, network topology, and trusted computing. They also raise the question of how to determine whether the protocols actually work, i.e., do they do what they are supposed to? To this end, they discuss a variety of formal methods approaches for verifying clock synchronization protocols, including interactive proofs, automated verification, and partial automation. Moreover, formal methods that can prove properties related to robustness remain quite challenging!

Overall, this article provides an interesting overview of robust clock synchronization, and points toward a variety of interesting open questions in the area. I hope that you enjoy the column!

The Distributed Computing Column is particularly interested in contributions that propose interesting new directions and summarize important open problems in areas of interest. If you would like to write such a column, please contact me.

Current Challenges in Reliable and Secure Clock Synchronization

Swen Jacobs and Christoph Lenzen
CISPA Helmholtz Center for Information Security

1 The Task

This article discusses reliable and secure clock synchronization, as well as its verification, with a focus on real-world application scenarios and open problems. Synchronizing clocks in a network $G = (V, E)$ is a fundamental task that has been studied since the inception of the field. The goal of a synchronization algorithm is to perpetually compute a *logical clock* L_v at each participating network node $v \in V$. The optimization criteria might vary with application. In this article, we focus on the following common choices:

- the *global skew* $\mathcal{G} = \sup_t\{\mathcal{G}(t)\}$, where $\mathcal{G}(t) = \max_{v,w\in V} |L_v(t) - L_w(t)|$ between any pair of nodes in the network;
- the *local skew* $\mathcal{L} = \sup_t\{\mathcal{L}(t)\}$, where $\mathcal{L}(t) = \max_{\{v,w\}\in E} |L_v(t) - L_w(t)|$ between any pair of neighbors in the network;
- bounds on the logical clock rates, i.e., $1 \leq \frac{dL_v}{dt}(t) \leq \alpha$ for some $\alpha > 1$.[1] In particular, it is not permitted to simply set all logical clocks to 0 forever or running them at exponentially decreasing rates.

Minimizing skew is challenging due to the inherent uncertainties in the system. Each node is equipped with a *hardware clock* H_v that approximates real time with a rate error of at most $\vartheta - 1$, i.e., for $t' > t$ it holds that[2]

$$t' - t \leq H_v(t') - H_v(t) \leq \vartheta(t' - t).$$

In addition, communication delay cannot be known precisely. To account for this, we assume that messages are under way for at least $d-u$ and at most d time, where

[1] For notational convenience, we normalize the minimum rate to 1.

[2] The error is one-sided to simplify notation. Because $\vartheta - 1 \ll 1$, this corresponds to $(1-\rho)(t' - t) \leq H_v(t') - H_v(t) \leq (1+\rho)(t' - t)$ for $\rho \approx (\vartheta - 1)/2$.

d is the maximum end-to-end *delay* and u the delay *uncertainty;* we assume that $(\vartheta - 1)d < u$.[3] For the sake of simplicity, we disregard heterogeneous systems in which the quality of clocks or links differs in this article, and pretend that all logical clocks can be initialized perfectly, i.e., $L_v(0) = 0$ for all nodes v.

The global and local skew bounds that can be achieved within this model have been identified to be $\Theta(uD)$ and $\Theta(u \log_{\alpha/(\vartheta-1)} D)$, respectively [10, 55], where D is the diameter of the network G. In this article, we discuss the challenges that arise when the theory underlying these results and approaches by practitioners face a reality in which faults and attacks are the norm. As will become clear, this leads to a whole range of new open problems, which require not only the techniques from distributed computing, but also cryptography and formal verification to evolve.

2 Why Do Faults and Attacks Matter?

Access to accurate time, being a basic service, is a crucial building block in many systems. This includes critical infrastructure, meaning that poor reliability or susceptability to attacks is an immense risk on a societal scale. To make this concrete, we now discuss several such systems and how they rely on a shared notion of time.

2.1 The Power Grid

The economic damage from even fairly short power outages is massive [12, 73, 75, 78]. At the same time, the power grid depends on microsecond accuracy in synchronizing monitoring devices to correlate measurements well enough to function; with increasing reliance on renewable energy sources, this becomes more and more important [25]. A failure of or successful attack on the timebase used by the power grid could cause global failure of the system, after which recovery will take at least several hours, cf. [27]. Attacks are viable and have been performed, with the synchronization subsystem being a viable attack vector [81, 59].

Note that the power grid is, inherently, a highly distributed system. Hence, it is virtually impossible to ensure that an attacker can access none of its components at all. This means that techniques dealing with worst-case, i.e., Byzantine, faults play a key role in securing it against attacks. As a convenient side effect, these can also increase the resilience of the system against faults that are not caused by attempted sabotage.

[3] This assumption is typically satisfied. If not, one can simulate hardware clocks with (up to a constant) this quality by bouncing messages back and forth along network links.

2.2 Cellular and Broadcast Networks

Cellular and broadcast networks require synchronization between cells to combat interference [21]; errors as small as microseconds can bring down entire networks [8]. Also here, the economic stakes of failure are high. Moreover, arguably our dependence on these networks also in times of crisis could render them critical infrastructure as well. In contrast to the power grid, for which central processing of measurements requires a small global skew, minimizing interference is a matter of minimizing the local skew.

2.3 Synchronization via the Internet

A standard, if inaccurate, way of obtaining time is to ask the Internet. Typically, this is done by querying time servers via the Network Time Protocol (NTP) [35, 65]. While fairly inaccurate, this method is popular due to requiring no additional resources on the client side. Thus, (too) many systems and services are likely to depend directly or indirectly on this approach.

As we discuss later, NTP and similar services are vulnerable to several attacks. Arguably, this means that critical services should not rely on this synchronization method. However, the sheer volume of NTP users means that improving reliability, security, and accuracy of Internet synchronization is worthwhile. It also suggests that it is likely that some crucial services will nonetheless be subject to NTP-based attacks.

2.4 Financial Sector

Banks are required to obtain "traceable" time with accuracy of 100 microseconds or better [64]. As suggested by this standard, the advantage of responding quicker to new information, even by milliseconds, provides a distinct advantage in high-frequency trading, cf. [14]. Equivalently, obtaining a timestamp "from the past" when committing a transaction yields the same advantage. From a theoretic point of view, the solution is to switch to a discretized, round-based market, in which trade requests are resolved based on reception times of requests at stock exchanges. However, such a change would require regulatory oversight to step in [15], which could necessitate a joint international initiative. In the current system, there is an incentive for the bank and its employees to manipulate timestamps for the sake of profit. Since timestamping occurs *within the bank's system,* it is, in principle, trivial to do so. As previous large-scale instances of fraud and malpractice [9, 85] clearly demonstrate, it is ill-advised to let the fox guard the hen house.

Note that this setting requires to rethink the time infrastructure having in mind that the *user* takes the role of the potential attacker. In contrast to the other examples, here it is insufficient to make sure that time is available and correctly recorded only at trustworthy nodes. On the other hand, accuracy requirements in the microsecond range render it challenging to directly involve remote parties in the timestamping procedure.

3 State of the Art

With the stage being set, let us revisit the state of the art and its limitations. As the examples in the previous section demonstrate in abundance, reliability and security are crucial in the wild, so this discussion will focus on these aspects.

3.1 Estimating Clock Offsets in Networks

Deployed solutions. This task is the larger part of what the Network Time Protocol (NTP) [36, 35] and the Precision Time Protocol (PTP) [71] seek to accomplish. The basic protocols are not concerned with security beyond message authentication. This is insufficient to cope with any mildly determined attacker, since verifying the content of a message does nothing to ensure its timing. In absence of bounds on communication delay, a man-in-the-middle attacker can arbitrarily and undetectably shift perceived relative time without needing to alter the content of messages [66]. There are works on improving resilience to some attacks for NTP (e.g. [41, 74]) and the possibility of using redundancy to increase the resilience of PTP has been considered as well [69]. However, what all of these works appear to have in common is that they consider the network to be given, whereas routing is either not considered or fixed by selecting a tree. Under these conditions, no substantial guarantees in face of faults and attacks are possible, and hence at best generic and vague statements are offered in this regard.

All-pairs estimation. From a theoretical angle, little has been published on the topic either. Of course one might endeavor to simulate full connectivity, as done in [29]. However, this approach has substantial drawbacks. In order to achieve any significant degree of resilience against an attacker taking control of network nodes or links, one must avoid that too many paths share the same edge or internal node. Apart from being difficult to realize in practical networks, this also means that one might be forced to prefer some longer paths to avoid relying too much on few nodes and edges. In turn, this can hurt the quality of measurements, as such longer paths will have increased cumulated delay uncertainty. This suggests non-trivial trade-offs between resilience and accuracy that so far have not been studied.

What is more, algorithms designed for a known given topology or specific classes of networks might perform better by not relying on offset measurements between all pairs of nodes. In other words, simulating full connectivity might harm robustness, security, or accuracy compared to topology-aware protocols. This conjecture is corroborated by some prior work that aims at achieving fault-tolerant synchronization in very sparse networks [16, 23].

Message authentication. Cryptographic authentication needs to play a major role for synchronization in networks. For many distributed tasks, such as consensus, it can be leveraged to increase resilience. This is also true for synchronizing clocks in fully connected systems [2]. Recall that d is the maximum end-to-end communication delay and u its uncertainty, i.e., the time between commencing message transmission and the receiver completing to process is between $d-u$ and d. It has been shown that the above resilience boost is also possible with asymptotically optimal skew of $O(u)$ [56]. However, there is a catch: this imposes the additional constraint that $d - u$ is a lower bound on the end-to-end delay on links with one *faulty* endpoint. Intuitively, this follows from the need to use indirect communication for authentication to be of value, together with the fact that faulty nodes cannot be detected reliably. This entails that malicious nodes must not get access to honest nodes' messages prematurely or deliver their own much faster than honest nodes; otherwise, they can enforce a decreased precision of time offset measurements, which in turn decreases the accuracy of synchronization that can be achieved. In contrast, the classic algorithm achieving the same asymptotic skew bound of $O(u)$ without authentication (and hence smaller resilience) [84] merely requires that the end-to-end delay bounds are satisfied on links between correct nodes.

Since any lower bound for a complete network extends to arbitrary networks, this points at potential fundamental obstacles to employing authentication for increased resilience, i.e., reducing requirements on network connectivity. In particular, one must avoid that an attacker can bypass the network or otherwise significantly speed up delivery of messages to or from nodes it controls.

3.2 Algorithms for Incomplete Networks

For complete networks, the asymptotically optimal skew of $O(u)$ can be achieved under optimal resilience to Byzantine (i.e., worst-case) faults [10, 24, 56, 84]. It is known that these guarantees can also be achieved in a self-stabilizing [22] manner with small stabilization time [42, 57, 58]. Similarly, one can simultaneously achieve asymptotically optimal skew between arbitrary pairs of nodes as well as neighbors [10, 55]. These results extend to crash faults in a straightforward way, and the algorithms can also be made self-stabilizing in asymptotically optimal time [54, Sec. 12.3]. In contrast, little is known about tolerance of non-benign faults or attacks in general networks.

Network augmentation. In [16], it is shown that one can augment a given network by replacing each node with a cluster of $\Theta(f)$ nodes and increasing to number of edges by factor $\Theta(f^2)$ to simulate the algorithm from [55] in a way tolerating up to f faults in each cluster. This is resource-efficient in the sense that if the original network was just barely connected, this overhead is necessary to handle f faults in each cluster. However, networks that are already connected well might need far fewer additional resources to enable us to achieve small skews in spite of faults. It should also be stressed that one might need even fewer resources when allowing for the possibility that also a small fraction of the nodes that faithfully execute the algorithm loses synchronization to the rest of the network. Note that this is not merely a question of connectivity, but also of how well-suited the available redundant paths are for synchronization, i.e., how long they are in the distance metric induced by communication delay uncertainties. Accordingly, so far none of these issues have been addressed in the literature.

3.3 Single Points of Failure in Deployed Systems

Radio communication. Global Navigation Satellite Systems (GNSS) are, wherever available, the most convenient source of accurate time. From a global point of view, the provider of such a system constitutes a single point of failure and can manipulate the time perceived by all receivers. A simple, very cost-effective method to reduce this risk is to rely on multiple GNSS services concurrently, selecting the median time value as reference. This means that a single bad actor cannot single-handedly change the perceived time. However, given the rather short list of available systems, the potential for collusion, and the need to receive a large number of satellite signals to implement this solution, it is not universally applicable.

Moreover, on the user's side, GNSS are fairly easy to jam or, by a more sophisticated attacker, to spoof or subject to delay attacks [82]. The use of multiple systems does not protect from such attacks, necessitating different means of obtaining or verifying the time.

Similar considerations apply to the terrestrial alternative of relying on long wave radio communication (e.g. DCF77 in Germany [31]), which furthermore is much less accurate than GNSS.

National Metrology Institutes (NMIs). Obtaining the time via the phone network or possibly a dedicated link from an NMI is another option to obtain fairly accurate time. However, relying exclusively on a single such reference again renders it a single point of failure. It is worth to note that, despite numerous checks and involvement of experts, there is no official standardization or documentation of the procedure NMIs use to obtain and adjust their time, and human involvement is no protection from bad actors. In fact, it is very plausible that, from a suitable position within an NMI, a single person could meddle with the time of everyone trusting in this NMI's time. In addition, also here an attacker might manipulate the time of a recipient or a group of recipients by delaying time messages from the NMI, without the need for altering these messages.

Lack of standardized redundancy. One might think that, obviously, this would prompt standardization of the use of multiple time references and/or means of obtaining time, with the goal to improve the reliability and security of deployed solutions. Unfortunately, nothing could be further from the truth. The examples with *best* behavior are possibly the most prominent network synchronization protocols, the Precision Time Protocol (PTP) and the Network Time Protocol (NTP), when taking into account additional literature. Regarding PTP, [69] briefly mentions the *possibility* to use redundant time references and routing paths to increase reliability and security, without any further discussion on requirements or best practices for doing so. Concerning NTP, Chronos makes an effort to leverage the availability of multiple time servers to increase resilience [74], but is called out for neglecting DNS-based attacks by [41]. Arguably, these works are efforts to plug individual holes in a very leaky bucket, since they focus on defending against certain attacks against a system and protocol that were not designed based on security considerations.

The user. As pointed out in the context of stock trades, it is possible that the user might want to manipulate their local time. We conclude that there is need for solving the task of forcing selfish or malicious parties to consistently report timing of operations. For this purpose, one option of interest is to use trusted computing technology, such as Trusted Platform Modules (TPM) [83] or Intel Software Guard Extensions (SGX) [3], at the client to prevent or at least substantially hamper abuse. These approaches establish trust anchors in hardware that should provide confidentiality and integrity of crucial operations, even in light of powerful attackers that can control the entire software stack or even have physical access to the system. However, these techniques cannot provide a trusted, reliable clock by themselves. Solutions like Intel SGX must either build on shared hardware resources between isolation domains, resulting in dependence on a potentially corrupted platform clock (e.g., if the attacker manipulates voltage and frequency scaling), or rely on higher privileged, untrusted software that can arbitrarily delay operations [3]. Even distinct coprocessors such as a TPM, which feature an internal clock, are no reliable time reference. A TPM depends on a properly managed platform, implying that an attacker with privileges on the system can set the TPM clock arbitrarily into the future, or make the TPM clock run 32.5% fast or slow [83]. In [6], the authors seek to secure the clock against manipulation, but their threat model does not account for attacks on the execution speed (e.g., by manipulating supply voltage or temperature), and the clock still has no guaranteed relation to UTC.

4 Does it *Actually* Work?

The development of clock-synchronization algorithms, as well as their implementation, are challenging and complex tasks. With this complexity comes the probability of human error. While designing algorithms, researchers will usually create hand-written proofs that supposedly witness correctness of their algorithms, but may contain errors. But even assuming correctness of these hand-written proofs, they only give us theoretical guarantees that not necessarily carry over to their real-world implementations. Therefore, it is crucial that the correctness of algorithms and their implementations is fully formalized and mechanically verified.

To obtain machine-checked formal proofs, there is a range of possibilities depending on the desired level of automation: *interactive* theorem provers allow us to formalize a very large range of systems and prove that they satisfy expressive formal specifications, but they require expert users and a large manual effort. In contrast, *automated* verification techniques alleviate the burden on the human designer, but are often restricted to certain types of systems, certain levels of abstraction, and certain properties to be proved. Between these two extremes there are all kinds of intermediate solutions, be it partial automation in interactive proofs, or manual efforts to massage a given system and specification until it fits into the fragment that is supported by an automatic method. In the following, we will first consider interactive tools, then fully automated ones, and finally techniques that try to get the best of both worlds by combining human guidance with automation. Regardless of the level of automation, we are interested in verification approaches that

1. can provide guarantees that hold regardless of the size of the network (called *parameterized verification* techniques)
2. have native support for faults and attacks, and
3. support real-time systems and properties.

While there is a lot of work on these aspects separately, or a combination of two of them (in particular on the verificiation of agreement protocols without real-time properties), will see that there are few existing approaches that support all of these features, and those that do exist are usually very restricted in some other aspect.

Interactive Mechanized Proofs. A first step towards formal correctness guarantees is a precise and testable specification of the intended functionality. In contrast to conventional design documents that contain prose or pseudocode without a testable semantics, a formal specification is precise, and this precision helps to eliminate ambiguities and clarify intention. Moreover, the formal specification can be gradually refined into an implementation, and can be checked for errors at any time during this process.

To support our intended applications, a tool for formalization needs to cover complex features, including concurrency, fault-tolerance, and time. A specialized formal language that already covers a lot of this is TLA$^+$ [52, 20, 51], which can be used to formally describe the set of all legal behaviors of a system, and is essentially based on set theory and temporal logic. While TLA$^+$ started as an academic tool, it has now also been used to support large-scale system development in industry, for example at Amazon Web Services [67, 68]. TLA$^+$ directly supports refinement of abstract specifications into more and more concrete algorithms and implementations. Correctness of an algorithm or an implementation then can either be shown by interactive mathematical reasoning in the TLA$^+$ proof system (TLAPS) [18], or by discharging certain types of proof obligations to SMT solvers [63] or model checkers [88, 46]. While TLA$^+$ does not explicitly support real-time properties, it has been argued that they can easily be modeled within the existing language, but automatically discharging the resulting proof obligations is a major challenge [53]. A number of languages and techniques have been inspired by TLA$^+$ and have similar strengths and weaknesses, for example the Ironfleet [76, 34] framework.

In addition to these specialized languages, more general interactive theorem provers have been used to reason about distributed systems. In particular, the PVS system has been used to verify certain textbook clock synchronization algorithms [77]. Moreover, Coq has been used as the basis of the Verdi framework for implementing and verifying distributed systems [86]. The focus of Verdi is on the use of verified system transformers, which simplify the task of reasoning about the correctness of refinements on the way from high-level specifications to low-level implementations. Like TLA$^+$ however, real-time properties are not directly supported. Another interactive theorem prover that has been used to develop specifications and proving correctness of implementations is HOL4 [79]. It has been used to verify low-level aspects like the network stack [11] or message queues [72], but it would take significant effort to scale such efforts to more complex systems.

Automated Verification. Automated verification techniques for distributed systems can be separated into those that give rigorous guarantees for systems with a parametric number of components, and those that under-approximate the possible behaviors of a distributed system by only considering a fixed number of components.

One of the most prominent tools is Alloy [39], which is not specialized to distributed systems, but allows to model infinite-state software systems in general. Alloy uses a *bounded* verification approach that makes verification decidable and can find many bugs, but does not give reliable correctness guarantees in general. Another tool with a similar behavior is MoDist [87]: given the code of a node in a distributed system, it instantiates it for a fixed number of processes and uses a model checking approach to check safety and liveness properties. Thus, errors that only manifest in systems with many processes will not be found, even if the search space for the chosen number of processes is explored exhaustively. However, MoDist has two notable features that many other approaches lack: first, it works directly on unmodified executable code, simulating the OS and the network, including failures such as message reordering and machine crashes. Second, it includes some support for real-time properties, in particular timeouts, by providing a *virtual clock* mechanism that approximates the behavior of a real clock, restricting its analysis to certain parts of the code and to simple comparisons against certain program variables.

A slightly different support for obtaining correct distributed systems is provided by Mace [43], which allows the designer to specify a distributed system in a restricted and structured domain-specific language, model check this high-level specification (with a special focus on liveness properties [44]), and compile it to a C++ implementation that inherits the desired properties and includes code for failure detection and handling. Like Alloy and MoDist, model checking is restricted to a fixed number of processes.

Another completely automated approach is implemented in MCMT, a model checker modulo theories [30]. The idea is to model distributed systems as infinite-state systems whose state variables are arrays (of unbounded length), and use SMT solvers to compute reachable sets of states. Since this reasoning naturally involves quantification, which is in general not supported in a complete way by SMT solvers, the technique relies on heuristics for quantifier instantiation that are tailored for the use case of model checking. The approach assumes that the system is given as an array-based transition system and does not provide support for automatic translation from executable code. On the other hand, it has been used to reason about distributed systems with a real-time component, such as different versions of the Fischer protocol [17]. An extension of the MCMT approach that expands its applicability and also includes an integration with a deductive verification framework has recently been introduced [19].

Finally, there are *parameterized model checking* techniques that are often restricted to a system model with certain types of communication or synchronization, but within such a fragment provide a decision procedure for properties that hold regardless of the number of components [13, 37, 38]. However, most of the existing results in this direction do not support strong attacker or fault models.

An exception is ByMC, the Byzantine Model Checker [47]: it is based on *threshold automata* that can model distributed protocols that count messages and make progress when a certain number of messages (the *threshold*) has been received, and it supports Byzantine faults in a number of nodes that is defined relative to the threshold. ByMC supports parametric verification of safety and liveness properties, i.e., a violation of the properties will be found regardless of how many processes are needed to exhibit the error, and if no violations are found, the system is provably correct for any number of processes [48]. While ByMC supports strong attackers, like most parameterized model checking approaches it does not support real-time properties.

There is a line of research in parameterized model checking that is able to give timing guarantees by modeling processes as timed automata [1], but in turn it does not support strong attacker models. However, an approach based on timed automata has recently been used to model and verify a basic gossiping clock synchronization protocol [80] (with explicit modeling of possible faulty behavior). In addition, there are approaches that support the verification of symbolic time bounds [40], but it is unclear if these can be extended to the verification of clock synchronization protocols.

Techniques with Partial Automation. Above, we mentioned the MoDist tool, which verifies an abstract algorithm and compiles it to an implementation that is guaranteed to inherit the desired properties. A more intricate variant of this approach is taken by the Civl verification framework [49, 50], which is based on an approach called layered refinement. The basic idea is that a proof of correctness does not relate an implementation to a specification in a single step, but in several refinement steps that abstract away details from the implementation, while preserving the properties that are necessary to prove the specification. In this case, the developer is responsible for designing the different refinement layers, while correctness of the refinements can then be fully automated (for a fixed number of processes). A similar approach is taken by the Armada language and tool [60], which additionally allows the developer to extend the library of proof strategies, potentially enabling the verification of a larger range of programs.

Another important recent development that merges the interactive and automated approaches is Ivy [62], a "multi-modal" verification tool that allows interactive TLA$^+$-style proofs, but also has a strong focus on automated verification in decidable fragments, which ensures its predictability and stability against small perturbations in the input.

In some of the approaches mentioned before, the hard part of verification is the identification of an inductive invariant that implies the desired safety property. The I4 approach [61] aims at automating the search for inductive invariants by efficiently identifying invariants on small instances of the system (based on symbolic model checking), and generalizing them to invariants that hold for the protocol in general, regardless of the number of processes. After generalization, the existing Ivy tool is used to check correctness of the invariant. Generalization itself is based on a number of heuristics that have proven fruitful for some applications, but may have to be extended for other cases. An extension of the approach with a special form of predicate abstraction (called syntax-guided abstraction-refinement) and word-level reasoning is implemented in the AVR tool [32]. This approach enables the verification of more complex systems, because it can automatically abstract from the domain complexity that is outside of the considered problem, for example by concentrating on control-flow details while abstracting from the processed data.

A variation of this approach is also implemented in SWISS [33]: instead of identifying invariants by model checking a small instance of the protocol, the approach relies on restrictions of the invariant itself. The idea is that, since protocols are designed by humans that have a correctness argument in mind, their correctness must be provable based on a relatively simple invariant. Therefore, search is restricted to a well-defined finite set of candidate invariants. The obvious drawback is that this restriction may be too strong, and a suitable invariant may not be found even if it exists.

Handling Faults. While some of the approaches above handle faults explicitly, most of them do not. If such an approach is used to verify distributed systems, there are two main ways to obtain correctness guarantees also in the presence of faults: either by proving that the program, as is, is fault-tolerant, or by *making* it fault-tolerant. An example of the first approach uses the regular model checking framework for verifying parameterized systems, together with a formal fault model, to completely automate verification in the presence of faults [28]. For the second approach, there exist approaches that use synthesis techniques to automatically modify existing protocols in order to ensure fault-tolerance, for example implemented in the FTSyn tool [26]. These tools support different types of faults and combinations of faults.

5 A Wish List

From the discussion above, we derive a number of specific challenges of interest to be tackled. They fall broadly in two categories: understanding clock synchronization under faults or attacks, and suitable tools for their verification. Naturally, a third crucial challenge is to actually apply these techniques in the context of the problem areas listed earlier, implementing, verifying, and deploying algorithms in the wild.

5.1 Open Problems for Synchronization under Faults

Estimating clock offsets in incomplete networks. The most basic ingredient of clock synchronization algorithms is a subroutine for estimating clock offsets between the nodes of the network. Between neighbors, this is simply done by direct communication. If algorithms need to compute such estimates between non-neighboring nodes, the best (worst-case) accuracy is achieved via communication along the shortest path with respect to edge weights given by the measurement error induced by communicating along each edge. However, when faults or corruption of internal nodes or edges on the communication path become a possibility, redundant use of (node or edge) disjoint paths can increase resilience. Unfortunately, the additional paths might provide less accurate measurements. Accordingly, we need to understand how to achieve the best possible tradeoff between accuracy and resilience. This prompts a large number of specific research questions:

- Given a fixed set of k disjoint paths between two network nodes and a target resilience f, suppose that for each path we know a worst-case bound on the accuracy when measuring the offset between the two nodes' clocks using this path, provided it is fault-free. What is the best worst-case accuracy that can be guaranteed when computing an estimate based on taking measurements from all k paths? Is there a strategy that is concurrently optimal for all values of f? If not, what are the trade-offs? Note that this problem can be used to model a client seeking to robustly obtain accurate time when multiple time servers with known communication paths of non-uniform accuracy are available, by introducing a virtual node with perfect clock. A generalized version considers the setting when not all paths are disjoint.

- Given a network with the above edge weights and a pair of nodes, how to best select k paths for a given target resilience f? Is there a uniform strategy that is good for all values of f?

- How do the answers to the above questions change if we consider multiple pairs of nodes, but allow for estimation to fail for some pairs (possibly as a function of f) entirely? A particularly important special case is the all-pairs setting, since this corresponds to simulating a complete network for the purpose of synchronization. Combining the outcome with analyses of the resilience of algorithms for complete networks to edge failures could yield improvements over the state of the art.

- Given a network and a budget for adding edges or nodes, how much can we improve the suitability of the network for the above tasks? Here, edge cost may vary depending on the uncertainty that should be guaranteed, this uncertainty could be a function of the physical network topology, or a mixture of both.
- Taking this one step further, what happens if we get to design the network from scratch? Which topologies are most suitable for the above tasks?

Damage mitigation. If faults or an attacker cannot be entirely prevented from causing a disruption, it is important to limit the impact on the functionality of the system. This is the underlying philosophy of guaranteeing synchronization at all non-faulty nodes or all but a small fraction of the non-faulty nodes. However, there are further options serving this purpose.

One such option that has been neglected in theoretical work on the subject is to harness the local clocks of nodes more effectively to mitigate the effects of temporary disruptions. If one (re-)designs algorithms to adjust the computed local output clocks only at small rates comparable to their inherent drift from UTC, even a prolonged attack temporarily compromising a majority of the network has limited impact on the time at uncompromised nodes. This can dramatically raise the cost of an attacker trying to achieve a network-wide disruption, and it could entirely prevent some faults (like the bug from [21]) from affecting the functionality of the system. Concretely, cesium standards – a type of atomic clock – are off by no more than a few nanoseconds per day. If an algorithm limits its corrections to the point where it amplifies this "natural" drift to no more than 10 nanoseconds per day, an attack would need to be maintained for several *months* to induce an error of a single microsecond. In the meantime, the attack can be detected and repelled, without causing any disruption to applications requiring microsecond (or worse) accuracy that obtain their time from a non-corrupted node.

Next, instead of trying to maintain synchronization at all times, an important and well-known complementary approach is to ensure automatic recovery from disruptions. Taking this approach to the extreme, one may ask for self-stabilization, i.e., automatic recovery to a consistent system state after arbitrary transient faults [22].

If this is too costly or makes the system more vulnerable in other regards, one might settle for automatic recovery under additional assumptions. For example, if the consequences of global system failure are so dramatic that human intervention will be triggered anyway and the potentially faster recovery due to self-stabilization provides no relevant advantage, it is justified to assume that the majority of the nodes remain synchronized. This, in turn, greatly simplifies recovery for nodes that undergo transient faults, as they merely need to resynchronize to the majority, cf. [45].

A promising candidate assumption is that even after the disruption, the synchronization error with respect to the reference time, say UTC, is bounded by some value B (which is unknown and possibly much larger than the error bound under nominal conditions). This is a natural outcome of leveraging local clocks to mitigate the impact of faults and attacks as described above. We then ask that the system converges back to nominal synchronization quality as soon as possible. Note that this results in a very appealing synergy: by bounding the impact of an attack of duration T on synchronization quality by $O((\vartheta - 1)T)$, recovery then might be possible within $O((\vartheta - 1)T)$ time, while simultaneously maintaining that the output clocks drift at rate $O(\vartheta - 1)$ relative to UTC. In other words, in addition to mitigating the impact of attacks and faults, we get the desirable property that the output clocks maintain rates that are close to 1 for free!

Gradient clock synchronization. Gradient clock synchronization (GCS) seeks, in addition to minimizing the worst-case skew between arbitrary pairs of nodes in the network, to minimize the worst-case skew between network neighbors. The latter can be kept as small as $\Theta(u \log D)$ [55], where D is the network diameter; this an exponentially smaller dependence on D than can be achieved for arbitrary pairs [10]. This makes GCS promising for settings where the skew between neighbors is what really matters. For example, when synchronizing broadcast transmitters or mobile phone cells, the goal is to avoid interference between close-by cells [21].

Accordingly, reliable GCS synchronization algorithms are of interest. Rather than requiring to add redundancy to the existing network as suggested in [16], one could exploit already existing redundancy in specific networks. For example, if we consider the power graph of a grid graph, i.e., connecting node (i, j) to all nodes (i', j') with $\max\{|i' - i|, |j' - j|\} \leq 1$, it is plausible that a variant of the algorithm from [55] could be devised that tolerates one fault in each neighborhood. If this is successful, it seems likely that the result can be generalized to similar graphs; also note that having the above graph as subgraph of the communication network is sufficient for running such an algorithm.

Moreover, the GCS algorithm from [55] allows us to bound the rate at which the output clocks can be adjusted by $O(\vartheta - 1)$. Therefore, it naturally lends itself to the damage mitigation mechanism outlined above. However, since the skew bound between neighbors is comparatively small and network cells will not all be equipped with cesium standards, the time for responding to an attack is smaller. Here, it is of interest to consider the following game. Suppose an attacker corrupts one or a several nodes in order to disrupt synchronization for additional nodes. Moreover, assume that (non-corrupted) nodes repeatedly and securely report the clock offsets they observe to their neighbors to a central authority,[4] and that the

[4] To keep in line with the goal of avoiding single points of failure, this "central" authority might

central authority can securely issue the command to logically delete links from the network. Is there a strategy that the authority can employ to prevent the attacker from causing disruption on a larger scale? Note that this is a difficult question, as an attacker could also attempt to coax the central authority into disconnecting other nodes, and the goal is to minimize the number of impacted nodes. Moreover, it is of interest to consider the same question for mobile attackers, which try to evade capture while seeking to disrupt synchronization. If there are good solutions to these tasks, one Where feasible, we will try to achieve this gold standard. can achieve accurate and reliable synchronization between neighbors in practical settings without adding substantial redundancy to existing networks.

Trusted timestamping. Another area of study are the accuracy and security guarantees that can be established for a trusted hardware providing a timestamping service that relates to UTC. Recall that existing solutions offer no such guarantee, since they do not rely on communication with devices outside of the control of the attacker. We argue that it is necessary to involve bidirectional communication with external parties to establish the veracity of any claims about timing. Otherwise, such an approach is doomed to fail due to an attacker having full control of *when* timing information reaches trusted hardware components.

Clearly, this necessitates to rely on cryptographic authentication techniques. However, this does not necessitate to trust a central server or authority, since time can be maintained collaboratively. Thus, this task blends in naturally with questions about secure distributed synchronization primitives.

5.2 Open Problems in Verification of Synchronization Protocols

As detailed before, there has been a lot of research into the verification of distributed protocols, including in particular agreement protocols. However, most of the techniques only support one or two of the three crucial features needed for the verification of practical clock synchronization protocols (parameterized verification, faults/attacks, real-time properties). Since the state of the art is rather far from being able to solve these problems we present, even partial solutions would be very much appreciated, e.g., proving *some* of the desired properties. Let us recapitulate some promising research directions, and detail the open problems they are connected to.

Automatic Methods for Clock Synchronization Protocols. There have been a number of efforts to verify clock synchronization protocols or other distributed protocols with real-time constraints, but these have been rather limited: As a completely automatic approach, Spalazzi and Spegni [80] have introduced a method

also be implemented via consensus of several monitors.

for parameterized model checking of gossiping clock synchronization protocols. This is a good starting point, but their model is limited to a certain form of gossiping communication, and has other strong restrictions that make it difficult or impossible to express practical protocols. Therefore, we believe that research into stronger fragments that still allow completely automatic forms of verification will be a fruitful research direction. Independent dimensions in which these results can be extended are (i) lifting restrictions within their framework of gossiping protocols, (ii) extending their approach to models with other communication/synchronization primitives, and (iii) extending their approach to support stronger fault models and (symbolic) cryptographic primitives.

Other Automatic Verification Methods. In addition, there are lots of completely automatic verification methods that give impressive results for certain classes of distributed protocols [47, 30], but to the best of our knowledge none of them supports parameterized verification with real-time constraints and strong fault models. Moreover, very few of them (e.g. [4]) support complex network topologies, which we have identified as being necessary for optimal robustness, security and accuracy in Section 3.1. Thus, while these techniques are certainly also worthwhile as a basis for the verification of clock synchronization protocols, on a superficial level the gap to supporting them is even bigger.

Semi-Automatic Methods. While the completely automatic methods above strictly need to be extended to handle clock synchronization protocols, the situation is a bit different with semi-automatic methods: many of them could, at least in theory, be used as they are to verify real-time constraints under strong fault models, as long as the user encodes the system and its desired properties in a specific way, and guides verification according to the capabilities of the underlying verification method. However, it is not difficult to imagine that this is a sub-optimal approach that puts most of the burden on the user, and is unlikely to succeed for any but the smallest examples. Therefore, such approaches also need specialized extensions to directly support the specification and verification of real-time constraints and strong fault models.

For example, in approaches based on TLA$^+$ [52, 51], both strong faults and real-time constraints can in theory be supported, but making its handling efficient and making verification scale to interesting protocols (or even implementations) is a challenging task. We conjecture that the addition of at least some dedicated support for such features, both in specification and in automated handling of verification conditions, would already make a large difference.

Verification of Implementations. While there are existing approaches that can formally verify abstract clock synchronization protocols (under strong restrictions), and there are approaches for the verification of *implementations of* distributed protocols (instead of high-level algorithms), we are not aware of any approach that combines both features, verifying actual implementations of clock synchronization protocols.

Some of the techniques mentioned in Section 4 work directly on implementations (e.g., in C code), others only verify the properties of a high-level model that may be implemented in different ways, making it necessary to prove that the desired properties are maintained by the implementation. Thus, even if we extend the existing techniques to support real-time properties and strong attackers or faults, we may have to additionally verify that implementations faithfully realize the respective abstract algorithms and inherit the desired properties. Some of the semi-automatic techniques support this naturally, for example the "layered" verification approaches based on TLA$^+$, or the layered refinement in Civl [49] and Armada [60]. Techniques like that, if extended to support the necessary features, could also be used to solve this problem for approaches that only verify the correctness of high-level algorithms.

For our use case, such approaches might benefit from research in adjacent areas that face similar problems, like the verification of implementations of security protocols [7], or of controllers for cyber-physical systems [70, 5]. In particular the latter area should provide important insights, as their high-level models have real-valued variables for time and other physical quantities, while implementations are usually restricted to some finite-precision abstractions of the real numbers. We face the same problem in implementations of clock synchronization algorithms, or in other distributed protocols with real-time constraints, for that matter.

References

[1] Parosh Aziz Abdulla and Bengt Jonsson. Verifying networks of timed processes (extended abstract). In *TACAS*, volume 1384 of *Lecture Notes in Computer Science*, pages 298–312. Springer, 1998.

[2] Ittai Abraham, Srinivas Devadas, Danny Dolev, Kartik Nayak, and Ling Ren. Synchronous Byzantine Agreement with Expected $O(1)$ Rounds, Expected $O(n^2)$ Communication, and Optimal Resilience. In *Financial Cryptography and Data Security (FC)*, pages 320–334, 2019.

[3] Fritz Alder, N. Asokan, Arseny Kurnikov, Andrew Paverd, and Michael Steiner. S-FaaS: Trustworthy and Accountable Function-as-a-Service Using Intel SGX. In *Cloud Computing Security Workshop (CCSW)*, pages 185–199, 2019.

[4] B. Aminof, S. Jacobs, A. Khalimov, and S. Rubin. Parameterized model checking of token-passing systems. In *VMCAI*, volume 8318 of *LNCS*, pages 262–281. Springer, 2014.

[5] Adolfo Anta, Rupak Majumdar, Indranil Saha, and Paulo Tabuada. Automatic verification of control system implementations. In *Proceedings of the Tenth ACM Inter-*

national Conference on Embedded Software, EMSOFT '10, page 9–18, New York, NY, USA, 2010. Association for Computing Machinery.

[6] Fatima M. Anwar, Luis Garcia, Xi Han, and Mani Srivastava. Securing Time in Untrusted Operating Systems with TimeSeal. In *Real-Time Systems Symposium (RTSS)*, pages 80–92, 2019.

[7] Matteo Avalle, Alfredo Pironti, and Riccardo Sisto. Formal verification of security protocol implementations: a survey. *Formal Aspects Comput.*, 26(1):99–123, 2014.

[8] Chris Baraniuk. UK Radio Disturbance Caused by Satellite Network Bug, 2016. https://www.bbc.com/news/technology-35463347.

[9] Victor A. Beker. *The American Financial Crisis*, pages 45–59. Springer International Publishing, 2016.

[10] Saâd Biaz and Jennifer L. Welch. Closed Form Bounds for Clock Synchronization under Simple Uncertainty Assumptions. *Information Processing Letters (IPL)*, 80(3):151—157, 2001.

[11] Steve Bishop, Matthew Fairbairn, Michael Norrish, Peter Sewell, Michael Smith, and Keith Wansbrough. Engineering with logic: HOL specification and symbolic-evaluation testing for TCP implementations. In *POPL*, pages 55–66. ACM, 2006.

[12] The Cost of Blackouts in Europe, 2016. https://cordis.europa.eu/article/id/126674-the-cost-of-blackouts-in-europe.

[13] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. *Decidability of Parameterized Verification*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2015.

[14] Eric Budish, Peter Cramton, and John Shim. The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response. *The Quarterly Journal of Economics (Q J Econ)*, 130(4):1547–1621, 2015.

[15] Eric Budish, Robin S. Lee, and John J. Shim. A Theory of Stock Exchange Competition and Innovation: Will the Market Fix the Market? *SSRN*, 2019.

[16] Johannes Bund, Christoph Lenzen, and Will Rosenbaum. Fault Tolerant Gradient Clock Synchronization. In *Symposium on Principles of Distributed Computing (PODC)*, pages 357–365, 2019.

[17] Alessandro Carioni, Silvio Ghilardi, and Silvio Ranise. MCMT in the land of parametrized timed automata. In *VERIFY@IJCAR*, volume 3 of *EPiC Series in Computing*, pages 47–64. EasyChair, 2010.

[18] Kaustuv Chaudhuri, Damien Doligez, Leslie Lamport, and Stephan Merz. The tla$^+$ proof system: Building a heterogeneous verification platform. In *ICTAC*, volume 6255 of *Lecture Notes in Computer Science*, page 44. Springer, 2010.

[19] Sylvain Conchon and Mattias Roux. Reasoning about universal cubes in MCMT. In *ICFEM*, volume 11852 of *Lecture Notes in Computer Science*, pages 270–285. Springer, 2019.

[20] Denis Cousineau, Damien Doligez, Leslie Lamport, Stephan Merz, Daniel Ricketts, and Hernán Vanzetto. TLA + proofs. In *FM*, volume 7436 of *Lecture Notes in Computer Science*, pages 147–154. Springer, 2012.

[21] Magnus Danielson. GPS Incident on Broadcast Networks. Technical report, U.S. Civil GPS Service Interface Committee (CGSIC), 2016. `https://rubidium.se/~magnus/papers/GPSincidentA6.pdf`.

[22] Edsger W. Dijkstra. Self-Stabilizing Systems in Spite of Distributed Control. *Communications of the ACM*, 17(11):643–644, 1974.

[23] Danny Dolev, Matthias Függer, Christoph Lenzen, Martin Perner, and Ulrich Schmid. HEX: Scaling Honeycombs is Easier than Scaling Clock Trees. *Journal of Computer and System Sciences (JCSS)*, 82(5):929–956, 2016.

[24] Danny Dolev, Joe Halpern, and H. Raymond Strong. On the Possibility and Impossibility of Achieving Clock Synchronization. In *Symposium on Theory of Computing (STOC)*, pages 504–511, 1984.

[25] Stephen Dominiak and Ulrich Dersch. Precise Time Synchronization of Phasor Measurement Units with Broadband Power Line Communications. Technical report, Swiss Federal Office of Energy SFOE, 2017.

[26] Ali Ebnenasir, Sandeep S. Kulkarni, and Anish Arora. Ftsyn: a framework for automatic synthesis of fault-tolerance. *Int. J. Softw. Tools Technol. Transf.*, 10(5):455–471, 2008.

[27] J.W. Feltes and Carlos Grande-Moran. Black Start Studies for System Restoration. In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–8, 2008.

[28] Dana Fisman, Orna Kupferman, and Yoad Lustig. On verifying fault tolerance of distributed protocols. In *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 315–331. Springer, 2008.

[29] Marc Frei, Jonghoon Kwon, Seyedali Tabaeiaghdaei, Marc Wyss, Christoph Lenzen, and Adrian Perrig. G-SINC: Global Synchronization Infrastructure for Network Clocks, 2022.

[30] Silvio Ghilardi and Silvio Ranise. MCMT: A model checker modulo theories. In *IJCAR*, volume 6173 of *Lecture Notes in Computer Science*, pages 22–29. Springer, 2010.

[31] Peter Glatzel. Timewarp DCF77-Testgenerator. *Elrad*, 2:88–91, 1996.

[32] Aman Goel and Karem A. Sakallah. AVR: abstractly verifying reachability. In *TACAS (1)*, volume 12078 of *Lecture Notes in Computer Science*, pages 413–422. Springer, 2020.

[33] Travis Hance, Marijn Heule, Ruben Martins, and Bryan Parno. Finding invariants of distributed systems: It's a small (enough) world after all. In *NSDI*, pages 115–131. USENIX Association, 2021.

[34] Chris Hawblitzel, Jon Howell, Manos Kapritsos, Jacob R. Lorch, Bryan Parno, Michael L. Roberts, Srinath T. V. Setty, and Brian Zill. Ironfleet: proving safety and liveness of practical distributed systems. *Commun. ACM*, 60(7):83–92, 2017.

[35] Internet Engineering Task Force (IETF). Network Time Protocol Version 4: Protocol and Algorithms Specification, 2010. `https://datatracker.ietf.org/doc/html/rfc5905`.

[36] Internet Engineering Task Force (IETF). Message Authentication Code for the Network Time Protocol, 2019. `https://datatracker.ietf.org/doc/html/rfc8573`.

[37] Nouraldin Jaber, Swen Jacobs, Christopher Wagner, Milind Kulkarni, and Roopsha Samanta. Parameterized verification of systems with global synchronization and guards. In *CAV (1)*, volume 12224 of *Lecture Notes in Computer Science*, pages 299–323. Springer, 2020.

[38] Nouraldin Jaber, Christopher Wagner, Swen Jacobs, Milind Kulkarni, and Roopsha Samanta. Quicksilver: modeling and parameterized verification for distributed agreement-based systems. *Proc. ACM Program. Lang.*, 5(OOPSLA):1–31, 2021.

[39] Daniel Jackson. *Software Abstractions - Logic, Language, and Analysis*. MIT Press, 2006.

[40] Swen Jacobs, Mouhammad Sakr, and Martin Zimmermann. Promptness and bounded fairness in concurrent and parameterized systems. In *VMCAI*, volume 11990 of *Lecture Notes in Computer Science*, pages 337–359. Springer, 2020.

[41] Philipp Jeitner, Haya Shulman, and Michael Waidner. The Impact of DNS Insecurity on Time. In *Conference on Dependable Systems and Networks (DSN)*, pages 266–277, 2020.

[42] Pankaj Khanchandani and Christoph Lenzen. Self-Stabilizing Byzantine Clock Synchronization with Optimal Precision. *Theory of Computing Systems (TOCS)*, 63(2):261–305, 2019.

[43] Charles Edwin Killian, James W. Anderson, Ryan Braud, Ranjit Jhala, and Amin Vahdat. Mace: language support for building distributed systems. In *PLDI*, pages 179–188. ACM, 2007.

[44] Charles Edwin Killian, James W. Anderson, Ranjit Jhala, and Amin Vahdat. Life, death, and the critical transition: Finding liveness bugs in systems code (awarded best paper). In *NSDI*. USENIX, 2007.

[45] Attila Kinali, Florian Huemer, and Christoph Lenzen. Fault-tolerant Clock Synchronization with High Precision. In *Symposium on VLSI (ISVLSI)*, 2016.

[46] Igor Konnov, Jure Kukovec, and Thanh-Hai Tran. TLA+ model checking made symbolic. *Proc. ACM Program. Lang.*, 3(OOPSLA):123:1–123:30, 2019.

[47] Igor Konnov and Josef Widder. Bymc: Byzantine model checker. In *ISoLA (3)*, volume 11246 of *Lecture Notes in Computer Science*, pages 327–342. Springer, 2018.

[48] Igor V. Konnov, Marijana Lazic, Helmut Veith, and Josef Widder. A short counterexample property for safety and liveness verification of fault-tolerant distributed algorithms. In *POPL*, pages 719–734. ACM, 2017.

[49] Bernhard Kragl and Shaz Qadeer. Layered concurrent programs. In *CAV (1)*, volume 10981 of *Lecture Notes in Computer Science*, pages 79–102. Springer, 2018.

[50] Bernhard Kragl and Shaz Qadeer. The civl verifier. In *FMCAD*, pages 143–152. IEEE, 2021.

[51] Markus Alexander Kuppe, Leslie Lamport, and Daniel Ricketts. The TLA+ toolbox. In *F-IDE@FM*, volume 310 of *EPTCS*, pages 50–62, 2019.

[52] Leslie Lamport. The temporal logic of actions. *ACM Trans. Program. Lang. Syst.*, 16(3):872–923, 1994.

[53] Leslie Lamport. Real-time model checking is really simple. In *CHARME*, volume 3725 of *Lecture Notes in Computer Science*, pages 162–175. Springer, 2005.

[54] Christoph Lenzen. Clock Synchronization and Adversarial Fault Tolerance, 2021. `https://www.mpi-inf.mpg.de/departments/algorithms-complexity/teaching/summer21/clock-synchronization-and-adversarial-fault-tolerance`.

[55] Christoph Lenzen, Thomas Locher, and Roger Wattenhofer. Tight Bounds for Clock Synchronization. *Journal of the ACM (JACM)*, 57(2), 2010.

[56] Christoph Lenzen and Julian Loss. Optimal Clock Synchronization with Signatures. *CoRR*, abs/2203.02553, 2022.

[57] Christoph Lenzen and Joel Rybicki. Near-Optimal Self-Stabilising Counting and Firing Squads. *Distributed Computing (DC)*, 32(4):339–360, 2019.

[58] Christoph Lenzen and Joel Rybicki. Self-Stabilising Byzantine Clock Synchronisation Is Almost as Easy as Consensus. *Journal of the ACM (JACM)*, 66(5):32:1–32:56, 2019.

[59] Yao Liu, Peng Ning, and Michael K. Reiter. False Data Injection Attacks against State Estimation in Electric Power Grids. *Transactions on Information and System Security*, 14(1), 2011.

[60] Jacob R. Lorch, Yixuan Chen, Manos Kapritsos, Bryan Parno, Shaz Qadeer, Upamanyu Sharma, James R. Wilcox, and Xueyuan Zhao. Armada: low-effort verification of high-performance concurrent programs. In *PLDI*, pages 197–210. ACM, 2020.

[61] Haojun Ma, Aman Goel, Jean-Baptiste Jeannin, Manos Kapritsos, Baris Kasikci, and Karem A. Sakallah. I4: incremental inference of inductive invariants for verification of distributed protocols. In *SOSP*, pages 370–384. ACM, 2019.

[62] Kenneth L. McMillan and Oded Padon. Ivy: A multi-modal verification tool for distributed algorithms. In *CAV (2)*, volume 12225 of *Lecture Notes in Computer Science*, pages 190–202. Springer, 2020.

[63] Stephan Merz and Hernán Vanzetto. Automatic verification of TLA + proof obligations with SMT solvers. In *LPAR*, volume 7180 of *Lecture Notes in Computer Science*, pages 289–303. Springer, 2012.

[64] Mifid ii, 2018. `https://www.esma.europa.eu/policy-rules/mifid-ii-and-mifir`.

[65] David L. Mills. Network Time Protocol (NTP), 1985. `https://www.hjp.at/(st_a)/doc/rfc/rfc958.html`.

[66] Lakshay Narula and Todd E. Humphreys. Requirements for Secure Clock Synchronization. *IEEE Journal of Selected Topics in Signal Processing (JSTSP)*, 12(4):749–762, 2018.

[67] Chris Newcombe. Why amazon chose TLA +. In *ABZ*, volume 8477 of *Lecture Notes in Computer Science*, pages 25–39. Springer, 2014.

[68] Chris Newcombe, Tim Rath, Fan Zhang, Bogdan Munteanu, Marc Brooker, and Michael Deardeuff. How amazon web services uses formal methods. *Commun. ACM*, 58(4):66–73, 2015.

[69] Karen O'Donoghue, Dieter Sibold, and Steffen Fries. New Security Mechanisms for Network Time Synchronization Protocols. In *Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS)*, pages 1–6, 2017.

[70] Junkil Park, Miroslav Pajic, Oleg Sokolsky, and Insup Lee. Automatic verification of finite precision implementations of linear controllers. In *TACAS (1)*, volume 10205 of *Lecture Notes in Computer Science*, pages 153–169, 2017.

[71] IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, 2020. IEEE Std 1588-2019 (Revision of IEEE Std 1588-2008).

[72] Tom Ridge. Verifying distributed systems: the operational approach. In *POPL*, pages 429–440. ACM, 2009.

[73] Alan H. Sanstad, Qianru Zhu, Benjamin Leibowicz, Peter H. Larsen, and Joseph H. Eto. Case Studies of the Economic Impacts of Power Interruptions and Damage to Electricity System Infrastructure from Extreme Events. Technical report, Berkeley Lab, 2020.

[74] Neta Rozen Schiff, Michael Schapira, Danny Dolev, and Omer Deutsch. Preventing (Network) Time Travel with Chronos. In *Applied Networking Research Workshop (ANRW)*, pages 17–31, 2018.

[75] Michael Schmidthaler and Johannes Reichl. Assessing the Socio-economic Effects of Power Outages ad hoc. *Computer Science - Research and Development*, 31:157–161, 2016.

[76] Fred B. Schneider. Technical perspective: Ironfleet simplifies proving safety and liveness properties. *Commun. ACM*, 60(7):82, 2017.

[77] Detlef Schwier and Friedrich W. von Henke. Mechanical verification of clock synchronization algorithms. In *FTRTFT*, volume 1486 of *Lecture Notes in Computer Science*, pages 262–271. Springer, 1998.

[78] M. Sforna and M. Delfanti. Overview of the Events and Causes of the 2003 Italian Blackout. In *Power Systems Conference and Exposition (PSCE)*, pages 301–308, 2006.

[79] Konrad Slind and Michael Norrish. A brief overview of HOL4. In *TPHOLs*, volume 5170 of *Lecture Notes in Computer Science*, pages 28–32. Springer, 2008.

[80] Luca Spalazzi and Francesco Spegni. Parameterized model checking of networks of timed automata with boolean guards. *Theor. Comput. Sci.*, 813:248–269, 2020.

[81] Chih-Che Sun, Adam Hahn, and Chen-Ching Liu. Cyber Security of a Power Grid: State-of-the-art. *International Journal of Electrical Power & Energy Systems*, 99:45–56, 2018.

[82] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the Requirements for Successful GPS Spoofing Attacks. In *Conference on Computer and Communications Security (CCS)*, pages 75–86, 2011.

[83] TPM Working Group. Trusted Platform Module Library Specification - Part 1: Architecture, 2019. `https://trustedcomputinggroup.org/wp-content/uploads/TPM-Rev-2.0-Part-1-Architecture-01.07-2014-03-13.pdf`.

[84] Jennifer L. Welch and Nancy A. Lynch. A New Fault-Tolerant Algorithm for Clock Synchronization. *Information and Computation (Inf Comput)*, 77(1):1–36, 1988.

[85] Duncan Wigan. Case Study: The Cum-Cum and Cum-Ex Schemes. Technical Report D4.5, Copenhagen Business School, 2019.

[86] James R. Wilcox, Doug Woos, Pavel Panchekha, Zachary Tatlock, Xi Wang, Michael D. Ernst, and Thomas E. Anderson. Verdi: a framework for implementing and formally verifying distributed systems. In *PLDI*, pages 357–368. ACM, 2015.

[87] Junfeng Yang, Tisheng Chen, Ming Wu, Zhilei Xu, Xuezheng Liu, Haoxiang Lin, Mao Yang, Fan Long, Lintao Zhang, and Lidong Zhou. MODIST: transparent model checking of unmodified distributed systems. In *NSDI*, pages 213–228. USENIX Association, 2009.

[88] Yuan Yu, Panagiotis Manolios, and Leslie Lamport. Model checking tla$^+$ specifications. In *CHARME*, volume 1703 of *Lecture Notes in Computer Science*, pages 54–66. Springer, 1999.

The Education Column

by

Juraj Hromkovič and Dennis Komm

ETH Zürich, Switzerland

juraj.hromkovic@inf.ethz.ch and dennis.komm@inf.ethz.ch

Testing of an Interactive Online Learning Environment with Focus on Algorithmic Tasks

Dario Naepfer
Department of Computer Science, ETH Zurich, Switzerland
dnaepfer@student.ethz.ch

Abstract

In this paper, we present a concept of computer science education, its implementation in an interactive online learning environment, and the extensive testing thereof. The goal is to support the development of algorithmic thinking by interactively solving small input instances of computing problems. The four approached competences are as follows:

1. To understand the abstract problem description and give proof of comprehension by classifying solution candidates into feasible and unfeasible solutions.
2. To find a solution for a given problem instance.
3. To find several different solutions for a given problem instance.
4. To apply criteria to evaluate and compare solutions, and to search for optimal solutions.

The contribution of this paper is twofold: First, we design the didactic approach and implement a learning environment in accordance with the aforementioned competences. Second, we test the environment in schools under various circumstances and present results of live testing, survey sessions including written feedback, as well as empirical test data derived from survey statistics. The empirical test covers user experience, intuitiveness of the learning environment, task difficulty, as well as satisfaction. This provides a genuine analysis of the application of the learning environment and allows drawing conclusions on the effectiveness of the applied didactic concept.

1 Introduction

Since the very beginning of humanity, Computer Science has been an important part of human culture due to its common roots with mathematics and written languages. This provides a good reason to consider informatics as a pivotal concept of education. Furthermore, it also qualifies the idea to include informatics as a fundamental pillar in the curricula of schools.

Multiple scientists contributed to the characterization of the discipline of informatics: Nygard [14] defined informatics as conceptual modelling and information systems. By Harel [8], informatics was depicted as a discipline covering computational, behavioural, and cognitive complexity, whereas Denning and Rosenblum [6] classified informatics as one of the four principal domains of science. Hromkovič and Lacher [12] further expanded the existing definitions as they added abstraction and symbolic representation, providing more efficiency. By describing the three roots of informatics, they presented a more holistic view of the discipline in the context of science than earlier descriptions, providing an overview of the potential of informatics education.

We define algorithmic thinking following Serafini [15]: the ability to systematically find solutions for problems with automated solution methods based on an iterative procedure. This makes it possible to extract, model, and represent information such that it can be written in terms of finite series of symbols chosen from a given set. Further, it characterizes the method, provides arguments supporting correctness, and analyses the amount of resources needed for execution. Also, it allows for a third party (e.g., a computer) to execute the solution method.

The learning environment developed here [13] is related to the concepts presented in *Computer Science Unplugged* [2,3], *"Abenteuer Informatik" (Computer Science Adventure)* [7], or *Algorithmic Adventures* [11]. Our environment is based on the concept of the textbook *"einfach Informatik 3/4" (Simply Computer Science)* [9, 10]. The authors present their as follows [5]:

> The starting point is merging constructionism and critical thinking. Constructionism with its "learning by doing" and "learning by getting things to work" enables designing a teaching process in which students acquire knowledge by creating products, analysing the properties and the functionality of their own products, and finally derive motivation to improve these products. Critical thinking asks us not to teach products of science and technology and their applications, but to teach the creative process of their development. To implement this approach, we use the historical method, allowing the students to learn by productive failures in the process of searching for a solution.

The concept presented in this paper approaches four competences. The students solve tasks for problem instances derived from common algorithmic problems, such as the *knapsack problem*, *vertex cover* or *dominating set*. Step by step, the student is asked to verify, find, evaluate, and finally optimize solutions for given problem instances. The tasks are designed for the third and fourth grade of elementary school implemented in an online learning environment based on the web framework *Vue.js* and were then exhaustively tested.

2 Didactic Goals

In this section, we present the didactic goals of the learning environment. The general didactic concept targeted aims to equip students to be able to encounter new problem formulations and learn how to solve them on their own. According to *Bloom's Taxonomy*, there are different levels of complexity depending on the cognitive dimensions of a task [1]. Here, four main competences are approached, which are based on the "Problem Solving and Algorithm Curriculum" that combines concepts such as constructivism and critical thinking with the hierarchy of *Bloom's revised taxonomy* [5]. These four approached competences are as follows:

1. To understand the abstract problem description and give proof of comprehension by classifying solution candidates into feasible and unfeasible solutions.

2. To find a solution for a given problem instance.

3. To find several different solutions for a given problem instance.

4. To apply criteria to evaluate and compare solutions, and to search for optimal solutions.

The task levels for each task set are designed accordingly: Starting with simple challenges that are classification tasks – the curriculum is designed such that the difficulty of the assignments improves gradually. Step by step, the students are taught the targeted algorithmic skills, similar to the classification informatics tasks of the *Bebras challenge on informatics* [4]. In this way, active learning and self-improvement are promoted. The students should learn to think in a way that enables them to approach new challenges and assignments with sophisticated steps and to solve them.

The tasks are designed such that the different levels of a curriculum look very similar. This creates recognition value, i.e., the user is familiarized with the problem and thus knows the general concept of the task when approaching the next difficulty level. Additionally, with multiple feasible solutions for problem instances, the student's creativity is required and challenged. The students learn to solve similar problems with yet different solutions or, in other words, learn how to apply algorithms. Random task instances further minimize the possibility of cheating or copying solutions. The students are compelled to find ways to solve the problems on their own, since there are no external sources or databases to help them. We will now link each task level with a competence or learning objective and describe each one in more detail.

Level 1: Understand Problems and Verify Solution Candidates

The first competence level approaches the student's capability to interpret a given problem instance description. He or she has to decide whether a given solution candidate is a feasible solution or not. In doing so, the student proves the competence of correct interpretation. A solution candidate is feasible if and only if it fully meets the task specifications. The tasks are modelled as decision problems, where the user must answer at least one question regarding the correctness of the proposition. The student is required to interpret the proposition of the problem instance correctly, fully understand the criteria, and then evaluate the given solution proposition accordingly. Furthermore, in some cases, the students learn to justify why a specific problem instance is or is not feasible.

Level 2: Find Solutions

At this competence level, the students are confronted with tasks for which a feasible solution must be found without any further help. The learning objective here is to work independently to find ways of dealing with a new challenge. This is more difficult than just verifying a solution candidate, since the student's creativity to come up with a new solution is required. The only condition a solution must fulfil is that it is valid. The quality of the solution according to a criterion is not taken into consideration here. All solutions are considered of equal value. One exercise type at this level has an intermediate version: the user is asked to complete a partial specification of a solution or instructed to find a solution on his or her own. In this case, however, the computer has already taken several steps towards a feasible solution candidate, so the number of valid solutions decreases significantly. If this eliminates some easy solutions, the difficulty level may change in comparison to an original level 2 task instance.

Level 3: List Multiple Solutions

The third competence level approaches the competence to find multiple solutions that differ from each other. In contrast to the previous level, it is less useful to use a mere brute-force search and more effective to develop a strategy to solve the problem. Students could later learn to use decision trees to systematically list all solutions as a new competence.

Level 4: Evaluate Solutions and Determine Optimal Solutions

Here, finding optimal solutions represents the highest competence level. It requires the student to evaluate different solutions with respect to a given criterion. This enables the student to compare the solutions and select an optimal one. Given the target group, it is evident that the number of possible solutions must be kept small such that a task instance can actually be solved. The variety of solutions must be small enough to enable the student to list all the different solutions, and to choose one of the best.

3 Test in Schools

In a final stage of the project, the learning environment was deployed on a test website and tested in various schools and grades. In this way, feedback for minor adaptions and improvements and empirical test data could be gathered. The test procedure was designed and arranged to include students from a wide range of origin, background, and academic performance to enable genuine and representative feedback to be gathered. In the following subsection, the test procedure will be examined and explained. The statistical results of the live testing feedback will then be presented in detail.

3.1 Test Extent

3.1.1 Test Locations and Levels

The testing phase was carried out in various schools and in different grades. In total, a variety of third and fourth grade classes as well as groups of especially talented students from second to eight grade completed the testing procedure. Further, online access was distributed to teachers from various places who agreed to conduct the live testing and to provide feedback. In total, over a hundred people tested the platform. The feedback was collected, examined, and evaluated in order to improve the learning environment.

3.1.2 Test Procedure

The test procedure was structured in the following way: First, information about the project, instructions, and guidelines was announced. Then, each participating student was provided with a computer or convertible, and was instructed to solve problem instances within a given time slot for each different task type. The students were encouraged to try to understand the exercise on their own and use the tutorial in case of problems. At the end of each time slot, the exercise type was changed. At the end of the test, a feedback round was conducted, and each student had to fill out an online feedback form to enable statistical data to be gathered.

3.2 Empirical Test Results

The feedback form to collect statistical data was designed the following way: First, general statistics were gathered such that the feedbacks could be categorized by grade, age, and school. The feedback form contained five main paragraphs, where the interviewee had to describe the perception of the learning environment. The paragraphs included questions on how interesting the tasks were, how much fun it was to solve them, whether the tasks were easy to understand, whether the tutorial was useful, and whether the number of exercises was satisfying. Some statistics are shown below to illustrate the feedback. For simplicity's and brevity's sake, only overall results are presented. The data includes representative feedback from about 70 students, randomly distributed in terms of origin, background, and academic performance.

First, the platform was evaluated in terms of understandability and intuitiveness. The participating students were asked how long it took them to understand the exercises, and whether they spent much time figuring out what the exact task description was. The statistical result of the survey is visualized in Figure 1. Over 90 percent of all students considered the tasks to be intuitive and easy to understand. Only a small percentage did not understand the tasks and had to ask someone else for help. These results line up with the design requirement that the learning environment should provide an intuitive and self-explanatory interface.

Further, how often the users did not understand tasks and then opened the tutorial was measured. Note that the tutorial contains a text and tutorial video that can be opened if the task is unclear. On the one hand, the data collected provides insight into the approaches users choose to handle new tasks. On the other hand, it yields information on the intuitiveness of each task. The result of this test is visualized in Figure 2 and aligns with the statistics laid out previously on intuitiveness of the tasks (see Figure 1).

Additionally, observations have shown that most of the students immediately tried to solve the task by testing and trying things out by themselves instead of

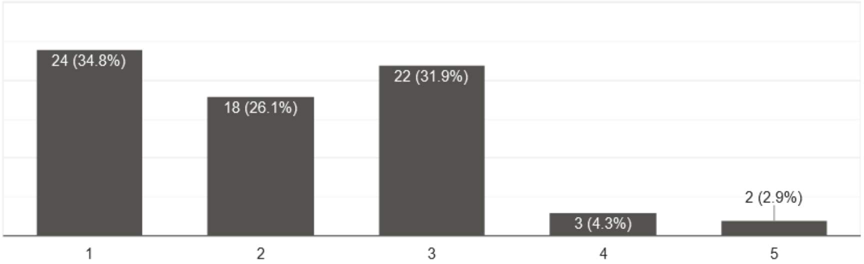


Figure 1: Diagram representing the task intuitiveness, linearly scaled from 1 to 5. 1 corresponds to 'very intuitive' and 5 to 'very unintuitive'.

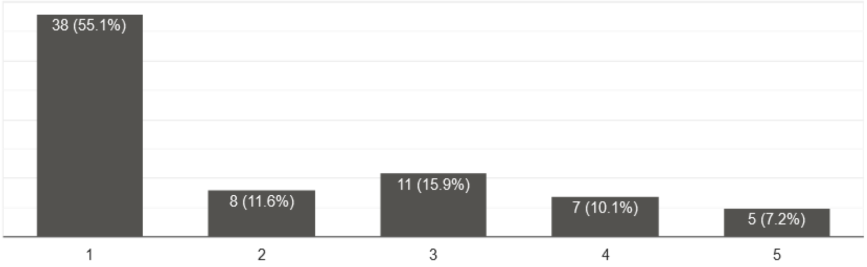


Figure 2: Diagram representing the consultations of an additional tutorial, linearly scaled from 1 to 5. 1 corresponds to 'not very often' and 5 to 'very often'.

reading the instructions. According to the collected data, more than half of the students tested did not even look at the tutorial once, and less than 20 percent opened it regularly when solving different tasks. This conclusion lines up with the feedback from testing staff, who noticed that some students did not even read the introductory sentence for each exercise, and immediately started clicking on the screen until they began to understand how to solve the respective task.

Next, statistics on user satisfaction were gathered. The user was asked whether he or she liked the exercises and was content with the way the tasks were constructed. The result of this survey is visualized in Figure 3 and matches the written feedback of the students. An overwhelming majority of the users indicated that they were pleased with the learning environment. Over 80 percent of the students liked the tasks or liked the tasks a lot. Only a small percentage did not like all the exercises (see leftmost bars in Figure 3). The written feedback analysis has shown that challenges that were perceived to be either too easy or too hard resulted in a low fun factor for a given task. These results agree with academic performance

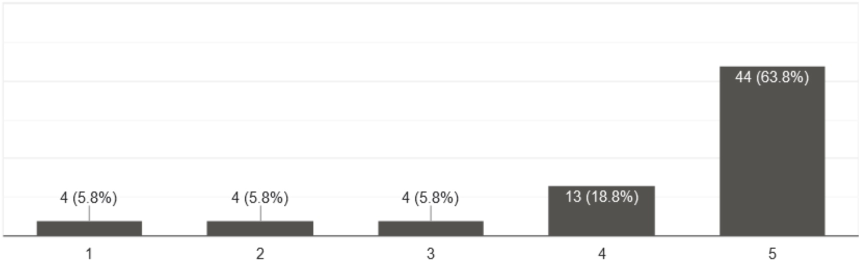


Figure 3: Diagram representing the perceived fun factor, linearly scaled from 1 to 5. 1 corresponds to ‘not so much fun’ and 5 to ‘much fun’.

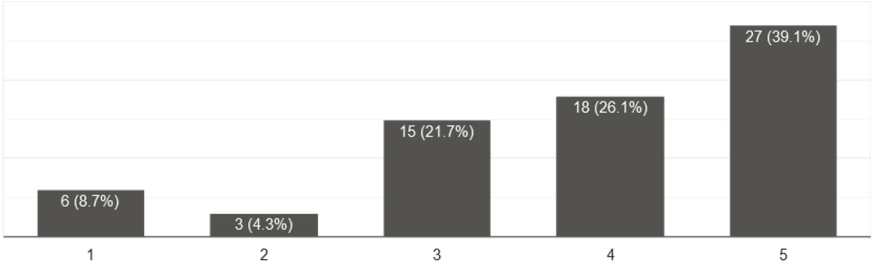


Figure 4: Diagram representing the perceived interest attached to the tasks, linearly scaled from 1 to 5. 1 corresponds to ‘not very interesting’ and 5 to ‘very interesting’.

statistics, which follow a normal distribution.

Further, an analysis on the attractiveness of the learning environment was performed. The tested students had to evaluate how interesting and exciting the tasks were. The visualization of this data set can be found in Figure 4. The result was rather positive, since nearly 90 percent of the students thought the tasks were at least more or less interesting, and overall nearly 40 percent rated the exercises as highly fascinating.

Subsequently, specific data to measure the difficulty of the tasks was collected. The users were asked how difficult the exercises were to solve, and whether the problems were challenging. Also, the time it took to solve single task instances was measured. Figure 5 gives a rough overview of the feedback data for this section. The results showed a normal distribution, as was expected. A small percentage of students was able to handle the tasks effortlessly, others were overwhelmed by the different task sets; and most students were seriously challenged. However, the statistics gathered suggest that slightly more students found the tasks to be on

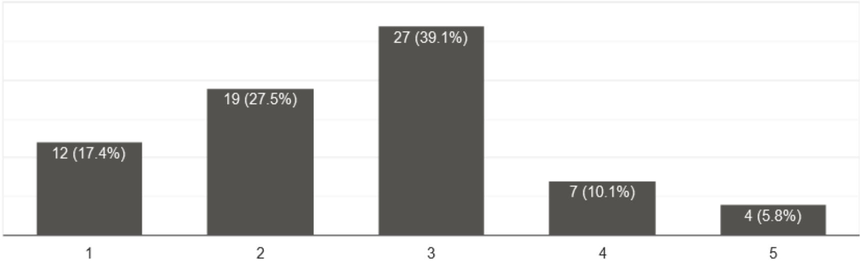


Figure 5: Diagram representing the perceived task difficulty, linearly scaled from 1 to 5. 1 corresponds to 'easy' and 5 to 'hard'.

the easy side within the normal distribution.

In the final evaluation section, the students were asked whether the number of exercises was sufficient. The result of this poll is visualized in Figure 6. About three out of four students were content with the variety of tasks.

Lastly, written feedback on specific questions was collected. The students were asked what they did or did not like about the platform, and what improvements they desired. The results of this questionnaire do not deviate from the previously presented data. In total, two out of three were highly positive: Over 20

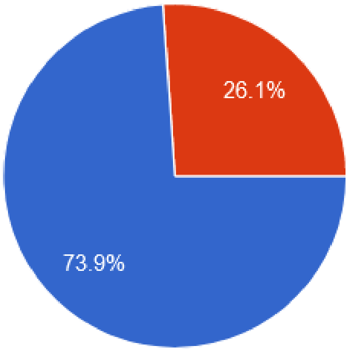


Figure 6: Diagram representing the perceived task variety. The fraction coloured in blue stands for 'good task variety', and the one coloured in red stands for 'rather small task variety'.

percent of those questioned explicitly remarked that they were pleased with the learning platform as a whole; 17 percent explicitly mentioned a specific task set on *Vertex Cover* as exceptionally interesting; 15 percent called the *Dominating Set* tasks well-designed. The tasks concerning the *Knapsack Problem* were mentioned in about 10 percent of the feedback forms. Furthermore, about 30 percent of the students explicitly indicated that they had no suggestions for improvements, while only a few individuals elaborated on ideas for enhancing the learning experience. Further, the feedback data supported the optimization of the learning environment in terms of user experience and performance. Linguistic vagueness could be clarified in several task descriptions, and additional explanations were added to improve understandability. Also, the environment could be further technically enhanced as some program parts were revised as a result of specific feedback. This concludes the feedback from the testing phase.

4 Conclusion and Discussion

The development of technical and didactic ways of teaching students is vital to ensure a sustainable education in the 21$^{\text{st}}$ century. Our contribution is to characterize an online learning environment that features diverse tasks that support the development of algorithmic thinking. The empirical test results indicate that the developed concept and derived tasks are user-friendly, intuitive, and interesting for the targeted student group. So far, the presented didactic concept has succeeded in the conducted tests. Everyone is invited to join the process by contributing and providing further feedback. Especially, new ideas for task concepts and test approaches are greatly appreciated.

Acknowledgements

First of all, I would like to thank Juraj Hromkovič for the opportunity to write this paper, the uncomplicated procedure, and feedback during the process. Also, I would like to express my gratitude to Dennis Komm, Regula Lacher, Giovanni Serafini, and Esther Baumgartner for the ongoing support and constructive feedback throughout the proceedings.

References

[1] Lorin W. Anderson and David R. Krathwohl: *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives.* 2nd edition. Longman, New York 2001.

[2] Tim Bell: Establishing a nationwide CS curriculum in New Zealand high schools: providing students, teachers, and parents with a better understanding of computer science and programming. *Communications of the ACM*, 57(2):28–30, 2014.

[3] Tim Bell, Jason Alexander, Isaac Freeman, and Mick Grimley: Computer science unplugged: school students doing real computing without computers. *New Zealand journal of Applied Computing and Information Technology* 13(1):20–29, 2009.

[4] Valentina Dagienė, Juraj Hromkovič, and Regula Lacher: A two-dimensional classification model for the Bebras tasks on informatics based simultaneously on subfields and competencies. In *Proc. of the 13th International Conference on Informatics in School (ISSEP 2020)*, LNCS 12518, pages 42–54, Springer, 2020.

[5] Valentina Dagienė, Juraj Hromkovič, and Regula Lacher: Designing informatics curriculum for K–12 education: from concepts to implementations. *Informatics in Education*, 20(3):7–15, 2021.

[6] Peter J. Denning and Paul Rosenbloom: The profession of IT – computing: the fourth great domain of science. *Communications of the ACM*, 52(9):27–29, 2009.

[7] Jens Gallenbacher: *Abenteuer Informatik: IT zum Anfassen für alle von 9 bis 99 – vom Navi bis Social Media* 4th edition. Springer, 2017.

[8] David Harel: *Algorithms – The Spirit of Computing.* Addison-Wesley, 1987.

[9] Heinz Hofer, Juraj Hromkovič, Regula Lacher, Pascal Lütscher, and Urs Wildeisen: *einfach Informatik 3/4: Programmieren und Rätsel lösen* (Textbook for Students). 1st edition. Klett und Balmer AG, 2021.

[10] Heinz Hofer, Juraj Hromkovič, Regula Lacher, Pascal Lütscher, and Urs Wildeisen: *einfach Informatik 3/4: Programmieren und Rätsel lösen* (Textbook for Teachers). 1st edition. Klett und Balmer AG, 2021.

[11] Juraj Hromkovič: *Algorithmic Adventures: From Knowledge to Magic.* Springer, 2009.

[12] Juraj Hromkovič and Regula Lacher: The computer science way of thinking in human history and consequences for the design of computer science curricula. In *Proc. of the 10th International Conference on Informatics in School (ISSEP 2017)*, LNCS 10696, pages 3–11, Springer, 2017.

[13] Dario Näpfer: *An Interactive Learning Environment.* `https://bit.ly/einfachInformatik34`. Department of Computer Science. ETH Zurich, 2021. Last accessed 19 Jul 2021.

[14] Kristen Nygaard: Program development as a social activity. In *Proc. of the IFIP 10th World Computer Congress*, pages 189–198, 1986.

[15] Giovanni Serafini: *The Benefits of Computer Science Education in Primary School: Promoting Algorithmic Thinking and Mathematical Problem Solving.* Manuscript in preparation. Department of Computer Science. ETH Zurich, 2022.

The Computational Complexity Column

by

Michal Koucký

Computer Science Institute, Charles University
Malostranské nám. 25, 118 00 Praha 1, Czech Republic
koucky@iuuk.mff.cuni.cz
https://iuuk.mff.cuni.cz/~koucky/

Recent Progress on Derandomizing Space-Bounded Computation

William M. Hoza[*]

Abstract

Is randomness ever necessary for space-efficient computation? It is commonly conjectured that L = BPL, meaning that halting decision algorithms can always be derandomized without increasing their space complexity by more than a constant factor. In the past few years (say, from 2017 to 2022), there has been some exciting progress toward proving this conjecture. Thanks to recent work, we have new pseudorandom generators (PRGs), new black-box derandomization algorithms (generalizations of PRGs), and new non-black-box derandomization algorithms. This article is a survey of these recent developments. We organize the underlying techniques into four overlapping themes:

1. The *iterated pseudorandom restrictions* framework for designing PRGs, especially PRGs for functions computable by arbitrary-order read-once branching programs.
2. The *inverse Laplacian* perspective on derandomizing BPL and the related concept of local consistency.
3. *Error reduction* procedures, including methods of designing low-error weighted pseudorandom generators (WPRGs).
4. The continued use of *spectral expander graphs* in this domain via the derandomized square operation and the Impagliazzo-Nisan-Wigderson PRG (STOC 1994).

We give an overview of these ideas and their applications, and we discuss the challenges ahead.

[*]Simons Institute for the Theory of Computing at the University of California, Berkeley. Email: williamhoza@berkeley.edu

1 Introduction

In an effort to solve problems as efficiently as possible, algorithm designers often introduce randomness into their algorithms. This paradigm is undoubtedly ingenious and beautiful. However, random bits can themselves be considered a computational "resource" that might be costly or unavailable. At best, randomization trades one type of inefficiency for another. We therefore want to distinguish between cases in which randomization gives an intrinsic advantage and cases in which algorithms can be derandomized with little to no penalty. In this article, we focus on the question of how randomization affects *space complexity*.

1.1 Randomized Space-Bounded Computation

Informally, $\mathsf{BPSPACE}(S)$ is everything that can be decided using randomness and $O(S)$ bits of space. More precisely, for a function $S : \mathbb{N} \to \mathbb{N}$, a language L is in $\mathsf{BPSPACE}(S)$ if there exists a Turing machine A with the following features.

1. The machine A has three tapes: a read-only input tape, a read-write work tape, and a read-once "random tape" that is initially filled with uniform random bits.

2. For every $N \in \mathbb{N}$,[1] every input $\sigma \in \{0,1\}^N$, and every assignment to the random tape $x \in \{0,1\}^\infty$, the machine touches at most $O(S(N))$ cells of the work tape and eventually halts, outputting a Boolean value $A(\sigma, x) \in \{0,1\}$.

3. For every input $\sigma \in \{0,1\}^*$, we have

$$\begin{aligned} \sigma \in L &\implies \Pr_x[A(\sigma, x) = 1] \geq 2/3 \\ \sigma \notin L &\implies \Pr_x[A(\sigma, x) = 1] \leq 1/3. \end{aligned}$$

Let us assume that $S \geq \log N$, so the machine has enough space to store a pointer to an arbitrary location in its input. Note that we assume that the algorithm halts for *every* assignment to the random tape (not merely with high probability). Using this assumption, one can show that the algorithm halts within $2^{O(S)}$ steps.[2] We use

[1] In this article, we use uppercase N to denote the length of the input to a space-bounded algorithm. We use lowercase n to denote the number of random bits that the algorithm uses.

[2] Historically, there was more early interest in the alternative "non-halting" model in which we merely require the algorithm to halt with high probability [33, 72, 73, 45, 13, 52, 69]. Indeed, in the older literature, notation along the lines of "$\mathsf{BPSPACE}(S)$" typically refers to the non-halting model, whereas the halting model is discussed using augmented notation such as "$\mathsf{BP_HSPACE}(S)$." Today, the halting model is standard.

BPL to denote BPSPACE($\log N$). The classes RSPACE(S) and RL are defined the same way, except that we only allow one-sided error.

These models were first studied by Aleliunas, Karp, Lipton, Lovász, and Rackoff [4] more than four decades ago. They presented a randomized algorithm showing that the undirected connectivity problem is in RL, and they asked whether L = RL. Today, the specific problem of undirected connectivity is indeed known to be in L, thanks to Reingold's famous algorithm [65] (the climax of a long sequence of papers studying the space complexity of undirected connectivity [4, 14, 10, 57, 59, 9, 77, 65, 68]). It is commonly believed that more generally L = RL = BPL. By a padding argument, if L = BPL, then DSPACE(S) = BPSPACE(S) for every space-constructible $S \geq \log N$.

Superficially, this sounds like the same frustrating story that pervades complexity theory. "We have been studying these important complexity classes for many decades, and at this point we think we know the relationship between them, but we don't know how to prove it." The same can be said regarding P vs. NP, or P vs. BPP, or L vs. P, or countless other fundamental problems.

However, there is a widespread feeling that *the* L *vs.* BPL *problem is different.* Compared to (say) the problem of proving P = BPP, there is a great deal of *optimism* about the possibility of unconditionally proving L = BPL. This optimism is sensible because the BPL model has a crucial weakness: the read-once random tape.

1.2 The Read-Once Assumption

In the definition of BPL, the machine A is only permitted to read each cell of the random tape a single time; the tape head can move right but not left. The motivation for this assumption is that we are modeling a problem-solving agent who has access to a single fair coin. The agent can see the outcome of only the most recent coin flip. If they want to know the outcome of a previous coin flip, they ought to have written it down at the time that it occurred (and paid for it in terms of space complexity).

As a consequence of the read-once assumption, the action of A on its random bits can be modeled by a polynomial-width standard-order *read-once branching program* (ROBP), defined below.

Definition 1 (Standard-order ROBPs)**.** A *width-w length-n standard-order ROBP* f is defined by a start state $v_0 \in [w]$, a sequence of n transition functions $f_1, \dots, f_n \colon [w] \times \{0, 1\} \to [w]$, and a set of accepting states $V_{\mathrm{acc}} \subseteq [w]$. An input $x \in \{0, 1\}^n$ determines a sequence of states $v_0, v_1, \dots, v_n \in [w]$ by the rule

$v_i = f_i(v_{i-1}, x_i)$ for $i > 0$. The output of the program is given by

$$f(x) = \begin{cases} 1 & \text{if } v_n \in V_{\text{acc}} \\ 0 & \text{if } v_n \notin V_{\text{acc}}. \end{cases} \tag{1}$$

Equivalently, we can think of f as a directed graph with vertices arranged in $n + 1$ layers, $V_0, \ldots, V_n$, where $|V_i| = w$. For $i < n$, each vertex $u \in V_i$ has two outgoing edges leading to V_{i+1}, one labeled 0 and the other labeled 1. There is a designated start vertex $v_0 \in V_0$, and there is a set of designated accepting vertices $V_{\text{acc}} \subseteq V_n$. An input $x \in \{0, 1\}^n$ is interpreted as a sequence of edge labels, identifying a path $(v_0, v_1, \ldots, v_n) \in V_0 \times V_1 \times \cdots \times V_n$. The output of the program is once again given by Equation (1).

The term "standard-order ROBP" is not standard. In typical papers on derandomizing space-bounded computation, standard-order ROBPs are simply called "ROBPs."[3] In this article, we include the modifier "standard-order" to emphasize that the program reads the input bits from left to right: first x_1, then x_2, then x_3, etc.

If A is a randomized, halting log-space algorithm and σ is an input of length N, then the function $f(x) \overset{\text{def}}{=} A(\sigma, x)$ can be computed by a width-n length-n standard-order ROBP for a suitable value $n = \text{poly}(N)$; each state of the program encodes a configuration of the machine A. An appealing approach to derandomizing A is to design a *pseudorandom generator* (PRG) that fools standard-order ROBPs.

Definition 2 (PRGs)**.** Let $\mathcal{F}$ be a class of functions $f\colon \{0, 1\}^n \to \{0, 1\}$, let X be a distribution over $\{0, 1\}^n$, and let $\varepsilon > 0$. We say that *X fools $\mathcal{F}$ with error ε* if for every $f \in \mathcal{F}$,

$$|\Pr[f(X) = 1] - \Pr[f(U_n) = 1]| \leq \varepsilon,$$

where U_n denotes the uniform distribution over $\{0, 1\}^n$. An *ε-PRG* for $\mathcal{F}$ is a function $G\colon \{0, 1\}^s \to \{0, 1\}^n$ such that $G(U_s)$ fools $\mathcal{F}$ with error ε. The value s is called the *seed length* of G.

If we could construct a PRG G that 0.1-fools width-n length-n standard-order ROBPs with seed length $O(\log n)$ and space complexity $O(\log n)$, then we could conclude that $\mathsf{L} = \mathsf{BPL}$, because we could deterministically estimate the acceptance probability of an algorithm A on an input σ to within ±0.1 by computing $A(\sigma, G(x))$ for every seed x.

[3] Standard-order ROBPs are also sometimes referred to as "ordered branching programs," "layered branching programs," or "sequential-access ROBPs."

1.2.1 PRGs and Lower Bounds

Some readers might have an intuition that says that designing unconditional PRGs is hopelessly difficult. This intuition is indeed sensible in many contexts. For example, consider the problem of designing a PRG that fools *general* size-n branching programs, i.e., programs that may read their input bits any number of times and in any order. Such a PRG $G\colon \{0,1\}^s \to \{0,1\}^n$ would induce a corresponding "hard function" $h\colon \{0,1\}^{s+1} \to \{0,1\}$ that cannot be computed by size-n branching programs.[4] If G is computable in space $O(s)$, then so is h. Therefore, the problem of designing explicit PRGs for general branching programs is even harder than the problem of proving *branching program lower bounds* for explicit functions. Perhaps someday our grandchildren will manage to prove optimal lower bounds for branching programs, but until that day, we should probably consider optimal PRGs for general branching programs to be out of reach.[5]

The good news is that the *read-once* assumption is an absolute game-changer. In the read-once setting, optimal lower bounds are already known. For example, using standard communication complexity arguments, one can show that every standard-order ROBP computing the function

$$h(x_1,\dots,x_{2n}) = x_1\cdot x_{n+1} \oplus x_2\cdot x_{n+2} \oplus \cdots \oplus x_n\cdot x_{2n}$$

has width $2^{\Omega(n)}$. To design optimal PRGs for standard-order ROBPs, we "merely" need to bridge the gap between lower bounds and PRGs.[6] There is no clear "barrier" preventing us from designing optimal PRGs for standard-order ROBPs. This is one of the reasons that a proof that $\mathsf{L} = \mathsf{BPL}$ seems vastly more attainable than, say, a proof that $\mathsf{P} = \mathsf{BPP}$.

1.2.2 Nisan's PRG and Beyond

So far, we do have several explicit PRGs that unconditionally fool standard-order ROBPs, but they do not achieve the optimal seed length. Most famously, Nisan designed an explicit PRG that ε-fools width-w length-n standard-order ROBPs with seed length $O(\log(wn/\varepsilon)\cdot\log n)$ [58]. The optimal seed length would be $\Theta(\log(wn/\varepsilon))$.

Admittedly, at this point it has been over three decades since Nisan's work [58], and we still do not have explicit PRGs for polynomial-width standard-order ROBPs with seed length better than Nisan's $O(\log^2 n)$ bound. However, an extensive body

[4] Specifically, h is the indicator function of the set $\{G(x)_{1\ldots s+1} : x \in \{0,1\}^s\}$.

[5] Currently, the best lower bound known is Nečiporuk's near-quadratic lower bound [56]. Explicit PRGs for size-n branching programs are known with a near-matching seed length of $\tilde{O}(\sqrt{n})$ [42, 38].

[6] Note that the Nisan-Wigderson reduction [60] does not work here, because it does not preserve the read-once property.

of research on the L vs. BPL problem has shown how to "go beyond" Nisan's work [58] in one sense or another. This rich and sophisticated literature is full of valuable insights that profoundly clarify the role of randomness in computing, even though the central questions remain open.

In the remainder of this article, we survey exciting progress that has been made on the L vs. BPL problem in just the past few years. (See Saks' survey [69] for an overview of older work.) We structure our discussion around four recurring technical themes: the *iterated pseudorandom restrictions* framework (Section 2), the *inverse Laplacian* perspective (Section 3), *error reduction* procedures (Section 4), and *expander graphs* (Section 5).

2 Iterated Pseudorandom Restrictions

2.1 Arbitrary-Order ROBPs

Nisan's classic PRG [58] suffers from a strange weakness. It turns out that permuting the output bits does not, in general, preserve the pseudorandomness property [78]. In other words, Nisan's PRG does not fool "arbitrary-order ROBPs." An *arbitrary-order ROBP* is defined just like a standard-order ROBP (Definition 1), except that instead of reading the input bits in the standard order $x_1, x_2, \dots, x_n$, it reads the input bits in the order $x_{\pi(1)}, x_{\pi(2)}, \dots, x_{\pi(n)}$ for some permutation $\pi \colon [n] \to [n]$.[7]

An interesting line of work has shown how to construct alternative PRGs for ROBPs that work even in the arbitrary-order setting [12, 76, 20, 50, 31]. We highlight a breakthrough paper by Forbes and Kelley [31]. Building on several earlier papers [66, 37, 20], Forbes and Kelley constructed two explicit PRGs for arbitrary-order ROBPs.

Theorem 1 (PRGs for arbitrary-order ROBPs [31])**.** *For every $w, n \in \mathbb{N}$ and $\varepsilon > 0$, there exist explicit ε-PRGs for width-w length-n arbitrary-order ROBPs with seed lengths*

$$O(\log(wn/\varepsilon) \cdot \log^2 n) \tag{2}$$

and

$$\tilde{O}(w \cdot \log(n/\varepsilon) \cdot \log n). \tag{3}$$

These seed lengths are only a little worse than Nisan's seed length [58], yet the PRGs fool a more powerful model.

For our main application (derandomizing BPL), it is no loss of generality to assume that the random bits are read in the standard order $x_1, x_2, x_3, \dots$, so why

[7] To be clear, these programs are still "oblivious," meaning that vertices in the same layer read the same input bit. Arbitrary-order ROBPs are also called "unordered ROBPs" or "unknown-order ROBPs."

study arbitrary-order ROBPs? One reason is that they capture other interesting models of computation such as read-once formulas [12, 32, 22, 28, 29, 30]. Another reason is that studying arbitrary-order ROBPs forces us to develop *new techniques* for fooling ROBPs. Indeed, the ideas underlying Forbes and Kelley's PRGs [31] are completely different than those underlying Nisan's PRG [58]. Forbes and Kelley's PRGs [31] are based on the framework of *iterated pseudorandom restrictions* – our first "theme."

2.2 Forbes-Kelley Restrictions

Ajtai and Wigderson introduced the iterated restrictions framework in the context of pseudorandomness for AC^0 circuits [3]. Much later, Gopalan, Meka, Reingold, Trevisan, and Vadhan brought the framework to the world of L vs. BPL [36]. The idea is as follows. Our goal is to sample a string $X \in \{0,1\}^n$ that fools some function of interest $f\colon \{0,1\}^n \to \{0,1\}$. Our first step is to design a pseudorandom *restriction* $X \in \{0,1,\star\}^n$, i.e., we pseudorandomly assign values to a *pseudorandom subset* of the variables. We ensure that X "preserves the expectation" of f, meaning that $X \circ U$ fools f, where $X \circ U$ denotes the string obtained by sampling X and then replacing each $\star$ with a fresh truly random bit. Intuitively, designing such an X is easier than designing a full PRG, because in the analysis, some helpful truly random bits (U) are sprinkled in among the pseudorandom bits.

After assigning values according to X, our remaining task is to fool the restricted function $f|_X$. Therefore, we repeat the process, i.e., we sample a restriction X' that preserves the expectation of $f|_X$. Iterating in this way, we assign values to more and more variables. Eventually, we have assigned values to all the variables and hence we have a full PRG.

Forbes and Kelley's primary contribution is to show how to accomplish the first step, i.e., how to sample a pseudorandom restriction that assigns values to many variables while preserving the expectation of every bounded-width arbitrary-order ROBP [31]. Indeed, they prove the following.

Theorem 2 (Restrictions for arbitrary-order ROBPs [31])**.** *Let $w, n \in \mathbb{N}$ and $\varepsilon > 0$, and let $k = 4\log(wn/\varepsilon)$. Let D and T be k-wise independent n-bit strings (with uniform marginals), let U be uniform random over $\{0,1\}^n$, and assume that D, T, and U are mutually independent. Then $D + (T \wedge U)$ fools width-w length-n arbitrary-order ROBPs with error ε, where $+$ denotes bitwise* XOR *and $\wedge$ denotes bitwise* AND.

The strings D and T define a restriction X by letting T indicate the $\star$ positions and using D to assign values to the non-$\star$ positions. The statement that X preserves the expectation of f is equivalent to the statement that $D + (T \wedge U)$ fools f. The way of thinking exemplified by the latter statement can be called the "pseudorandomness plus noise" perspective [37, 49].

Using standard constructions of k-wise independent random variables [44], one can explicitly sample D and T using $O(k \log n) = O(\log(wn/\varepsilon) \cdot \log n)$ truly random bits. In expectation, each restriction assigns values to half of the living variables, so after roughly $\log n$ iterations, we should intuitively expect that all the variables have been assigned values. Indeed, a more careful argument shows how to achieve an overall seed length of $O(\log(wn/\varepsilon) \cdot \log^2 n)$ (see Forbes and Kelley's work for details [31, Section 7]).

The proof of Theorem 2 is a beautiful application of Boolean Fourier analysis. Forbes and Kelley's techniques [31] work particularly well in the constant-width setting. By leveraging "Fourier growth bounds" for ROBPs [66, 76, 20, 48], Forbes and Kelley obtain restrictions for constant-width arbitrary-order ROBPs with better parameters. In the constant-width case, rather than k-wise independent distributions, Forbes and Kelley use "δ-biased distributions," i.e., distributions that fool parity functions with error $\delta/2$ [55].

Theorem 3 (Restrictions for constant-width arbitrary-order ROBPs [31])**.** *Let $w \in \mathbb{N}$ be a constant. For every $n \in \mathbb{N}$ and $\varepsilon > 0$, there exists a value $\delta = \exp\left(-\tilde{O}(\log(n/\varepsilon))\right)$ such that the following holds. Let D and T be δ-biased random variables distributed over $\{0,1\}^n$, let U be uniform random over $\{0,1\}^n$, and assume that D, T, and U are mutually independent. Then $D + (T \wedge U)$ fools width-w length-n arbitrary-order ROBPs with error ε, where $+$ denotes bitwise* XOR *and $\wedge$ denotes bitwise* AND.

Using standard constructions of δ-biased distributions [55, 5], the random variables D and T of Theorem 3 can be sampled explicitly using $O(\log(n/\delta)) = \tilde{O}(\log(n/\varepsilon))$ truly random bits. This leads to a PRG for constant-width arbitrary-order ROBPs with seed length $\tilde{O}(\log(n/\varepsilon) \cdot \log n)$.

2.3 The Early Termination Technique

Forbes and Kelley's PRGs [31] are examples of restrictions-based PRGs with seed length polylog(n), similar to the seed length of Nisan's PRG [58]. In some cases, we can use the iterated restrictions framework to get seed lengths as low as $\tilde{O}(\log n)$ or even $O(\log n)$. The key idea is to show that after applying a few pseudorandom restrictions (say, poly($\log \log n$) many), the function f that we are trying to fool "simplifies" in some sense with high probability. When this occurs, we can *terminate the restriction process early*, and use some other approach to fool the restricted function, taking advantage of its simplicity.

This "early termination" technique was introduced by Gopalan, Meka, Reingold, Trevisan, and Vadhan [36], and it has turned out to be useful for quite a few PRG problems [36, 50, 28, 47, 49, 29, 30]. Let us briefly discuss three examples.

- Gopalan, Meka, Reingold, Trevisan, and Vadhan designed an explicit PRG for *read-once* CNF *formulas* with near-optimal seed length $\tilde{O}(\log(n/\varepsilon))$ [36].
- Doron, Hatami, and Hoza designed an explicit PRG for *read-once* AC^0 *formulas* with near-optimal seed length $\tilde{O}(\log(n/\varepsilon))$ [28].
- Doron, Meka, Reingold, Tal, and Vadhan designed an explicit PRG for constant-width arbitrary-order *monotone ROBPs* (defined next) with near-optimal seed length $\tilde{O}(\log(n/\varepsilon))$ [30].

Definition 3 (Monotone ROBPs)**.** Let f be a width-w length-n arbitrary-order ROBP with transition functions $f_1, \dots, f_n \colon [w] \times \{0,1\} \to [w]$. We say that f is *monotone* if, for each $i \in [n]$ and each bit $b \in \{0,1\}$, the transition function $f_i(\cdot, b)$ is a monotone function $[w] \to [w]$ [51, 30].

It turns out that constant-width arbitrary-order monotone ROBPs can simulate read-once AC^0 formulas [22, 30]. In turn, obviously read-once AC^0 formulas generalize read-once CNF formulas. Thus, the classes fooled by the three PRGs mentioned above form a hierarchy:

$$\begin{aligned}&\text{read-once } \mathsf{CNF}\text{s}\\ &\quad\subseteq \text{read-once } \mathsf{AC}^0\\ &\qquad\subseteq \text{constant-width arbitrary-order monotone ROBPs.}\end{aligned}$$

Over time, we are gradually figuring out how to fool *more and more powerful classes* with near-optimal seed length, building our way up toward the class of general (arbitrary-order) ROBPs. This type of progress (steadily improving the class of functions fooled) has turned out to be more feasible than insisting on fooling *all* (standard-order) ROBPs and trying to improve the seed length.

Recall that Forbes and Kelley's work (Theorem 3) shows how to assign values to half the input variables of a constant-width arbitrary-order ROBP at a cost of only $\tilde{O}(\log(n/\varepsilon))$ truly random bits. To get a full PRG in the monotone case, Doron, Meka, Reingold, Tal, and Vadhan show that after a few Forbes-Kelley restrictions, monotone ROBPs are likely to simplify [30]. Roughly speaking, the notion of simplification is that the *width* of the program steadily decreases until the function is trivial.

We remark that Doron, Meka, Reingold, Tal, and Vadhan's work [30] is one example where *techniques designed for the arbitrary-order case have turned out to* *be useful even for the standard-order case.*[8] This demonstrates the counterintuitive wisdom of working on problems that are even *more* difficult than the problems that we care about most.

2.4 A Challenge: Parity Gates

The iterated restrictions paradigm is flexible and powerful, especially when it is combined with the early termination technique. All of the recent work using these techniques is certainly exciting and encouraging. Unfortunately, however, we still do not have a clear path toward fooling all constant-width ROBPs (let alone polynomial-width ROBPs) with near-logarithmic seed length. Indeed, it seems that this line of work is perhaps "running out of steam."

To understand the limitations of these techniques, observe that for any arbitrary-order ROBP $f\colon \{0,1\}^n \to \{0,1\}$, we can define a more complicated function $g\colon \{0,1\}^{n^2} \to \{0,1\}$ by block-composing with the parity function, i.e.,

$$g(x_{11}, \dots, x_{nn}) = f\left(\bigoplus_{i=1}^{n} x_{1i}, \bigoplus_{i=1}^{n} x_{2i}, \dots, \bigoplus_{i=1}^{n} x_{ni}\right).$$

If the initial ROBP f has width $w = O(1)$, then g can be computed by an arbitrary-order ROBP of width $2w = O(1)$, but the early termination technique seems to break down when we try to apply it to g. It seems that (pseudo)random restrictions have very little effect on g, because a restriction of the parity function is always either the parity function or its complement. Fooling a typical restriction of g is thus at least as difficult as fooling f.

More concretely, consider the problem of fooling read-once AC^0 formulas *with parity gates* (Figure 1). Doron, Hatami, and Hoza gave an explicit PRG for this class with seed length $\tilde{O}(t + \log(n/\varepsilon))$ where t is the number of parity gates in the formula [28]. For the depth-2 case, we have explicit PRGs with near-optimal seed length [49, 50, 47], and in fact with "partially optimal" seed length $O(\log n) + \tilde{O}(\log(1/\varepsilon))$ [29]. However, when the depth is a large constant and the number of parity gates is unbounded, it seems quite difficult to achieve seed length $\tilde{O}(\log n)$.

3 The Inverse Laplacian Perspective

In light of challenges such as that discussed in Section 2.4, it is worthwhile to take a step back and ask whether we truly need to design better PRGs for ROBPs. After

[8] The monotone ROBP model was first introduced by Meka and Zuckerman [51], who were not concerned with issues of variable ordering. They presented PRGs for the standard-order case [51]; in the constant-width regime, their seed length matches Nisan's [58].

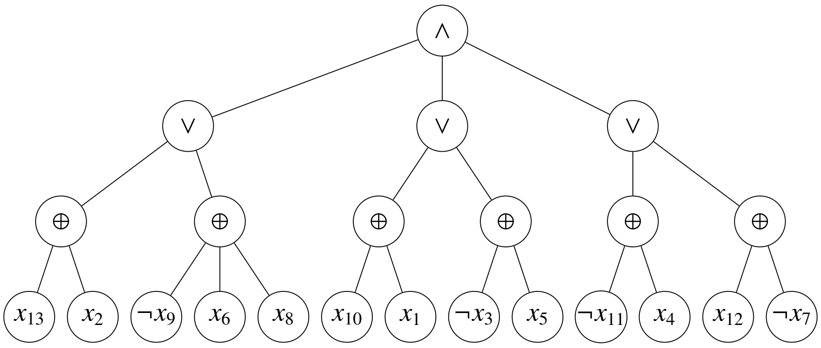


Figure 1: We would like to design explicit PRGs for constant-width (arbitary-order) ROBPs with near-optimal seed length $\tilde{O}(\log(n/\varepsilon))$. The class of read-once AC^0 formulas with parity gates is a challenging special case. Indeed, the case of read-once $\mathsf{AND} \circ \mathsf{OR} \circ \mathsf{PARITY}$ formulas already seems formidable.

all, our primary goal is derandomizing space-bounded computation. In this section, we discuss a non-PRG-based approach to proving $\mathsf{L} = \mathsf{BPL}$.

3.1 The Matrix of Expectations of Subprograms

To derandomize BPL, it suffices to design a deterministic log-space algorithm that is given a width-n length-n standard-order ROBP f and estimates $\mathbb{E}[f]$ to within a small additive error. There is no need to treat f as a black box; it is permissible to inspect the transitions of f and try to thereby gain some advantage. Since we are only concerned with space complexity, if we intend to estimate the expectation of the program, we might as well estimate the expectations of all subprograms, too.

Definition 4 (Subprograms)**.** Suppose f is a width-w length-n standard-order ROBP with layers $V_0, \dots, V_n$. Let $u \in V_i$ and $v \in V_j$ be vertices with $i \leq j$. We define the *subprogram* $f_{u \to v}$ to be the width-w length-$(j - i)$ standard-order ROBP on layers $V_i, V_{i+1}, \dots, V_j$ obtained from f by designating u as the start vertex and v as the unique accepting vertex.

Let us collect all the expectations of these subprograms $\mathbb{E}[f_{u \to v}]$ in an $m \times m$ matrix P, where m is the number of vertices in f, namely $m = w \cdot (n + 1)$. That is, for every pair of vertices u, v in f, if $u \in V_i$ and $v \in V_j$, then

$$P_{u,v} = \begin{cases} \mathbb{E}[f_{u \to v}] & \text{if } i \leq j \\ 0 & \text{if } i > j. \end{cases} \tag{4}$$

The following problem is essentially complete for BPL:

- **Input:** A width-n length-n standard-order ROBP f.
- **Output:** A matrix $\hat{P}$ that approximates the matrix of expectations of subprograms (P) to within additive entrywise error 0.1.

(By "essentially complete for BPL," we mean that a *decision version* of the problem is complete for the *promise version* of BPL with respect to deterministic log-space reductions. These technicalities do not seem to be important.)

3.2 The Inverse Laplacian of a Standard-Order ROBP

To try to approximate P, we can start by computing the *random walk matrix* W. By definition, for each pair of vertices u, v in f, the entry $W_{u,v}$ gives the probability of arriving at v when we start at u and take a single random step. Computing W is trivial: $W_{u,v}$ is half the number of edges from u to v.

The expectations of subprograms of f correspond to *powers* of W. Indeed, $(W^t)_{u,v}$ is the probability that a t-step random walk from u arrives at v. Therefore, if $u \in V_i$ and $v \in V_j$, then

$$(W^t)_{u,v} = \begin{cases} \mathbb{E}[f_{u\to v}] & \text{if } j - i = t \\ 0 & \text{otherwise.} \end{cases}$$

Consequently, there is a simple formula for the matrix of expectations of subprograms (P) in terms of the random walk matrix (W):

$$P = W^0 + W^1 + W^2 + \cdots + W^n. \tag{5}$$

Furthermore, $W^{n+1} = 0$, so we can simplify Equation (5) using the geometric series formula:

$$P = (I - W)^{-1}.$$

The matrix $I - W$ is called the (directed) *Laplacian matrix* of the program f and denoted L. Thus, we are looking for an *approximate inverse Laplacian* $\hat{P} \approx L^{-1}$. This way of thinking – the "inverse Laplacian perspective" – was introduced most clearly in work by Ahmadinejad, Kelner, Murtagh, Peebles, Sidford, and Vadhan [1], and it is our second technical "theme."

3.3 Local Consistency

A key benefit of the inverse Laplacian perspective is that it suggests a new way of thinking about error. Suppose that someone gives us a candidate matrix $\hat{P}$. Is $\hat{P}$ a good approximation to P? We cannot directly compare the entries of $\hat{P}$ to those of P, because we do not know P (remember, approximating P is essentially BPL-complete). However, we *can* compute the error *after multiplying by the Laplacian matrix*. That is, we can compare $\hat{P}L$ to the identity matrix. Define E to be the error matrix $E = I - \hat{P}L$.

This error matrix E has a natural probabilistic interpretation. Expanding the definition, we have $E = I - \hat{P} \cdot (I - W) = I + \hat{P}W - \hat{P}$. Therefore, if $u \in V_i$ and $v \in V_j$ where $i < j$, then

$$E_{u,v} = (\hat{P}W)_{u,v} - \hat{P}_{u,v} = \underbrace{\left(\sum_{s \in V_{j-1}} \hat{P}_{u,s} \cdot W_{s,v} \right)}_{(*)} - \hat{P}_{u,v}.$$

The entry $E_{u,v}$ measures the difference between two different methods of using $\hat{P}$ to estimate $\mathbb{E}[f_{u \to v}]$. The first method is to simply consult the (u, v) entry of $\hat{P}$, since after all $\hat{P}$ is intended to be an approximation to P. The second method is to look at $\hat{P}$'s estimates for the probabilities of arriving at vertices in the layer V_{j-1} that *precedes* v, and then propagate those probabilities forward by a single step, leading to quantity $(*)$.

Thus, E measures the extent to which $\hat{P}$ is locally *consistent with itself*; we refer to E as the matrix of *local consistency errors*. The term "local consistency" was introduced by Cheng and Hoza [23]; the connection between local consistency and the Laplacian matrix was observed by subsequent papers [24, 63, 39]. We will discuss an application of the notion of local consistency next. For additional applications of the inverse Laplacian perspective, see Sections 4 and 5.

3.4 One-Sided vs. Two-Sided Derandomization

Cheng and Hoza used the concept of local consistency to prove a new conditional derandomization of BPL [23]. Recall that a *hitting set generator* (HSG) is a one-sided version of a PRG.

Definition 5 (HSGs)**.** Let $\mathcal{F}$ be a class of functions $f\colon \{0,1\}^n \to \{0,1\}$ and let $\varepsilon > 0$. An *ε-HSG* for $\mathcal{F}$ is a function $G\colon \{0,1\}^s \to \{0,1\}^n$ such that for every $f \in \mathcal{F}$,

$$\text{if } \Pr[f(U_n) = 1] \geq \varepsilon, \text{ then there exists } x \text{ such that } f(G(x)) = 1.$$

If G is an ε-PRG for $\mathcal{F}$, then G is also is an ε'-HSG for $\mathcal{F}$ for every $\varepsilon' > \varepsilon$. HSGs are potentially much easier to construct than PRGs, so it is worthwhile to ask, what would be the applications of optimal explicit HSGs? Working through the definitions, one can easily show that an optimal explicit HSG for standard-order ROBPs would imply L = RL (one-sided derandomization). Cheng and

Hoza showed that it would also imply the stronger statement $\mathsf{L} = \mathsf{BPL}$ (two-sided derandomization) [23].

Theorem 4 (HSGs would derandomize BPL [23])**.** *Assume that for every $n \in \mathbb{N}$, there is a $\frac{1}{2}$-HSG for width-n length-n standard-order ROBPs that has seed length $O(\log n)$ and that is computable in space $O(\log n)$. Then $\mathsf{L} = \mathsf{BPL}$.*

Let us briefly sketch the proof of Theorem 4. Suppose we are given a width-n length-n standard-order ROBP f. Let G be an HSG with output length n^c, where c is a large enough constant. For each seed x, we think of $G(x)$ as a long stream of random bits and use it to compute a matrix $\hat{P}^{(x)}$ that is a candidate approximation to the matrix P of expectations of subprograms of f. Using the hitting property of G, one can show that there is at least one "good seed" x such that $\hat{P}^{(x)} \approx P$. To identify such a seed algorithmically, we find an x such that $\hat{P}^{(x)}$ has good local consistency.

We remark that an analogous theorem for time-bounded derandomization has been known for decades [6, 7, 18, 34]. In fact, Buhrman and Fortnow showed generically that derandomizing the promise version of RP would imply $\mathsf{P} = \mathsf{BPP}$, regardless of whether the derandomization is via an HSG [18]. An interesting open problem is to prove the analogous theorem for the space-bounded setting, generalizing Theorem 4.

4 Error Reduction Procedures

In the previous section, we introduced the inverse Laplacian perspective, and we discussed one application (the derandomization of BPL using a hypothetical HSG). There are several other applications of the inverse Laplacian perspective. These other applications take advantage of the rich literature on *fast, randomized* algorithms for approximately solving Laplacian systems of equations, starting with Spielman and Teng's seminal work [74]. Most especially, these other applications work by importing *error reduction* techniques – our third "theme" – to the space-bounded derandomization setting.

4.1 Non-Black-Box Error Reduction

As our first example, let us discuss a theorem by Ahmadinejad, Kelner, Murtagh, Peebles, Sidford, and Vadhan [1] (strengthening prior work by Hoza and Zuckerman [41]). Their theorem shows how to generically decrease the error of space-bounded derandomization algorithms. In the following, think of ε as negligibly small compared to n, such as perhaps $\varepsilon = 2^{-\operatorname{polylog}(n)}$.

Theorem 5 (Error reduction for non-black-box derandomization [1])**.** *Let $S : \mathbb{N} \to \mathbb{N}$ be a function. Assume that given a width-n length-n standard-order ROBP f, it is possible to deterministically compute $\mathbb{E}[f]$ to within $\pm 1/n^3$ in space $S(n)$. Then given f and $\varepsilon > 0$, it is possible to deterministically compute $\mathbb{E}[f]$ to within $\pm\varepsilon$ in space*

$$O(S(n) + \log n \cdot \log\log_n(1/\varepsilon)).$$

Let us sketch the proof of Theorem 5, which uses the inverse Laplacian perspective. Let P be the matrix of expectations of subprograms of f. Using the given $S(n)$-space algorithm, we can construct a matrix $\hat{P}$ such that $\left\|P - \hat{P}\right\|_\infty \leq O(1/n)$.[9] Let W be the random walk matrix of f, let $L = I - W$ be the Laplacian matrix, and let $E = I - \hat{P}L$ be the error matrix after multiplying by L (aka the matrix of local consistency errors). Then, we define a new approximation matrix $\hat{P}'$ by the formula

$$\hat{P}' = \hat{P} + E\hat{P} + E^2\hat{P} + \cdots + E^m\hat{P}$$

for a suitably chosen parameter m. (Intuitively, we start with $\hat{P}$, and then we add a sequence of finer and finer "correction terms" $E\hat{P}, E^2\hat{P}, \dots, E^m\hat{P}$.) Let us measure the quality of this new approximation. The key, again, is to measure quality *after* multiplying by the Laplacian matrix, which causes a telescoping sum:

$$\hat{P}'L = (I - E) + E \cdot (I - E) + E^2 \cdot (I - E) + \cdots + E^m \cdot (I - E) = I - E^m.$$

Amazingly, we have managed to replace E with E^m, which intuitively should mean that the errors are getting much smaller. This technique for decreasing the error of an approximate matrix inverse is called *preconditioned Richardson iteration*.

Ultimately, what we care about is entrywise closeness to P. We can bound the entrywise errors using the submultiplicative $\|\cdot\|_\infty$ matrix norm:

$$\begin{aligned}\left\|\hat{P}' - P\right\|_\infty = \left\|\left(\hat{P}'L - I\right) \cdot P\right\|_\infty = \|E^m \cdot P\|_\infty &\leq \|E\|_\infty^m \cdot \|P\|_\infty \\ &= \left\|\left(P - \hat{P}\right) \cdot L\right\|_\infty^m \cdot \|P\|_\infty \\ &\leq \left(\left\|P - \hat{P}\right\|_\infty \cdot \|L\|_\infty\right)^m \cdot \|P\|_\infty \\ &\leq O(1/n)^m \cdot O(n),\end{aligned}$$

which is at most ε if we choose a suitable value $m = O(\log(1/\varepsilon)/\log n)$. One can compute $\hat{P}'$ deterministically in space $O(S(n) + \log n \cdot \log m)$, completing the proof of Theorem 5.

[9] Indeed, $\|P - \hat{P}\|_\infty \overset{\text{def}}{=} \max_u \sum_v |P_{u,v} - \hat{P}_{u,v}| \leq n \cdot (n+1) \cdot \max_{u,v} |P_{u,v} - \hat{P}_{u,v}| \leq n \cdot (n+1) \cdot n^{-3}$.

4.2 Weighted Pseudorandom Generators (WPRGs)

The parameters of Theorem 5 are impressive; we pay very little penalty for error reduction, even when the target error ε is extremely small. The algorithm of Theorem 5 is non-black-box, because we must inspect the graph structure of the given ROBP to compute the matrices W, L, E, etc.

As discussed previously, non-black-box algorithms are sufficient for proving $\mathsf{L} = \mathsf{BPL}$. However, black-box algorithms are stronger, and they tend to be more useful as building blocks inside larger algorithms. What is the best way to compute $\mathbb{E}[f]$ to within a tiny additive error ε if we only have *query* access to a standard-order ROBP f? An ε-PRG clearly suffices for this task, but could there be an easier approach? This motivates the intriguing concept of a *weighted pseudorandom generator* (WPRG), introduced by Braverman, Cohen, and Garg [15].

Definition 6 (WPRGs)**.** Let $\mathcal{F}$ be a class of functions $f\colon \{0,1\}^n \to \{0,1\}$ and let $\varepsilon > 0$. An *ε-WPRG* for $\mathcal{F}$ is a pair (G, ρ), where $G\colon \{0,1\}^s \to \{0,1\}^n$ and $\rho\colon \{0,1\}^s \to \mathbb{R}$, such that for every $f \in \mathcal{F}$,

$$\left| \mathop{\mathbb{E}}_{x \sim U_n}[f(x)] - \mathop{\mathbb{E}}_{x \sim U_s}[f(G(x)) \cdot \rho(x)] \right| \leq \varepsilon. \tag{6}$$

A PRG is the special case $\rho \equiv 1$. Crucially, Definition 6 allows for $\rho(x) < 0$, which opens the door for the possibility of *error cancellation* in Equation (6).[10] One can think of these negative weights as effectively introducing a kind of "negative probability" into the picture; WPRGs are also called *pseudorandom pseudodistribution generators*.[11]

One can show that if (G, ρ) is an ε-WPRG for $\mathcal{F}$, then G is an ε'-HSG for $\mathcal{F}$ for every $\varepsilon' > \varepsilon$. Thus, we have a hierarchy,

$$\text{PRG} \implies \text{WPRG} \implies \text{HSG}.$$

When they introduced the concept of a WPRG, Braverman, Cohen, and Garg presented an explicit construction of an ε-WPRG for polynomial-width standard-order ROBPs [15] with seed length

$$\tilde{O}(\log^2 n + \log(1/\varepsilon)). \tag{7}$$

For comparison, recall that Nisan's PRG ε-fools polynomial-width standard-order ROBPs with seed length $O(\log^2 n + \log n \cdot \log(1/\varepsilon))$ [58]. Thus, Braverman, Cohen,

[10] WPRGs with nonnegative weight functions $\rho\colon \{0,1\}^s \to [0,\infty)$ are essentially equivalent to unweighted PRGs [63, Appendix C of ECCC version].

[11] Braverman, Cohen, and Garg coined the term "pseudorandom pseudodistribution" [15]. The alternative term "weighted pseudorandom generator" was introduced later, by Cohen, Doron, Renard, Sberlo, and Ta-Shma [24].

and Garg's seed length [15] is superior when ε is very small (again, the case $\varepsilon = 2^{-\operatorname{polylog}(n)}$ is good to have in mind). Prior to their work [15], it was not even known how to construct an ε-HSG with the seed length that they achieve.

Braverman, Cohen, and Garg's work [15] is quite complex. This spurred a search for simpler approaches [41, 21, 24, 63, 39]. In addition to achieving improved simplicity, this line of work was also able to remove the lower-order terms hiding under the $\tilde{O}$ in Equation (7).

Theorem 6 (Optimal-error WPRGs [39])**.** *For every $w, n \in \mathbb{N}$ and $\varepsilon > 0$, there is an explicit ε-WPRG for width-w length-n standard-order ROBPs with seed length $O(\log(wn) \cdot \log n + \log(1/\varepsilon))$.*

To prove Theorem 6, we start with Nisan's PRG with error $1/\operatorname{poly}(nw)$ and seed length $O(\log(wn) \cdot \log n)$. Then, we use the *preconditioned Richardson iteration* technique that we discussed in Section 4.1 to decrease the error of the PRG. Implementing this technique is not completely straightforward, because we are in the black-box setting, and hence we can no longer compute the matrices W, L, E, etc. However, two independent papers (one by Cohen, Doron, Renard, Sberlo, and Ta-Shma [24] and the other by Pyne and Vadhan [63]) contributed the insight that one can set up the WPRG *construction* in such a way that preconditioned Richardson iteration happens in the *analysis*. Finally, to achieve the seed length of Theorem 6, we combine these ideas with a suitable sampler trick [39].

In general, starting from an explicit PRG for width-w length-n standard-order ROBPs with error $1/(wn)^c$ and seed length s (for a suitable constant $c > 1$), we get an explicit WPRG for such programs with arbitrarily small error ε and seed length $O(s + \log(1/\varepsilon))$ [39]. There are other, related error reduction procedures that achieve slightly better parameters in some cases [41, 24, 63]. For example, consider standard-order ROBPs of width w and length $\log^c w$ for a constant $c \in \mathbb{N}$. Nisan and Zuckerman showed how to fool these short, wide programs with seed length $O(\log w)$ and a relatively large error such as $2^{-(\log w)^{0.99}}$ [61]. By applying an error-reduction procedure to the Nisan-Zuckerman PRG [61], Hoza and Zuckerman designed an explicit ε-HSG for these programs with asymptotically optimal seed length $O(\log(w/\varepsilon))$, even when ε is small [41]. It remains an interesting open problem to match this seed length with a WPRG.

4.3 Improving the Saks-Zhou Algorithm

Let us now discuss an application of low-error WPRGs. Recall our original derandomization goal: we want to deterministically decide languages in $\mathsf{BPSPACE}(S)$, for $S \geq \log N$, using as little space as possible.

Savitch's theorem [71] implies that $\mathsf{RSPACE}(S) \subseteq \mathsf{DSPACE}(S^2)$. The more general inclusion $\mathsf{BPSPACE}(S) \subseteq \mathsf{DSPACE}(S^2)$ follows from early work on the non-halting version of $\mathsf{BPSPACE}(S)$ [13, 45]. Later, Saks and Zhou used Nisan's PRG [58] in a sophisticated way to prove $\mathsf{BPSPACE}(S) \subseteq \mathsf{DSPACE}(S^{3/2})$ [70]. Now, decades later, we can finally improve Saks and Zhou's bound.

Theorem 7 (Improved derandomization of $\mathsf{BPSPACE}$ [39])**.** *Let $S : \mathbb{N} \to \mathbb{N}$ be a function satisfying $S(N) \geq \log N$. Then*

$$\mathsf{BPSPACE}(S) \subseteq \mathsf{DSPACE}\left(\frac{S^{3/2}}{\sqrt{\log S}}\right). \tag{8}$$

Admittedly, the bound of Equation (8) is only barely better than Saks and Zhou's $O(S^{3/2})$ bound [70]. Still, Theorem 7 potentially has some "psychological" value, because it demonstrates that Saks and Zhou's result [70] is not the "end of the road." There is no particular reason to think that Theorem 7 is the end of the road either. No compelling barriers to further progress are known; humanity has no real excuse for having not yet proven $\mathsf{L} = \mathsf{BPL}$.

The starting point for proving Theorem 7 is work by Armoni from more than two decades ago [8]. Armoni designed an explicit ε-PRG for width-w length-n standard-order ROBPs based on a generalization of Nisan and Zuckerman's techniques [61]. Armoni's seed length is slightly better than Nisan's seed length [58] in the regime $n \ll w$ and $\varepsilon \gg 1/w$ [8]. By combining his PRG with recent error reduction techniques [24, 63], we get an explicit WPRG with a seed length that is slightly better than Nisan's seed length [58] in the regime $n \ll w$, *even for low error* such as $\varepsilon = 1/\operatorname{poly}(w)$.

The original Saks-Zhou algorithm [70] uses Nisan's PRG with parameters in this regime ($n \ll w$ and $\varepsilon = 1/\operatorname{poly}(w)$) as a subroutine. Armoni showed how to use a generic PRG in place of Nisan's PRG [8], and Chattopadhyay and Liao showed more generally how to use WPRGs [21], building on an earlier suggestion by Braverman, Cohen, and Garg [15]. Combining these results proves Theorem 7. (See Figure 2.) This argument appears in work by Hoza [39], but to be clear, the ingredients all come from prior work [70, 8, 21, 24, 63]. Hoza's contribution to the proof of Theorem 7 is merely to put the pieces together [39].

Cohen, Doron, and Sberlo recently designed an algorithm that improves on Saks and Zhou's work [70] in a different direction [25]. Consider the following natural computational problem.

- **Input:** A value $n \in \mathbb{N}$ and a stochastic matrix $M \in \mathbb{R}^{w \times w}$, where each entry has bit complexity $O(\log(wn))$.
- **Output:** A matrix that approximates M^n to additive entrywise error 0.1.

When we restrict to the case $n = w$, the problem above is essentially complete for BPL. One can think of the Saks-Zhou algorithm as a method of solving the problem

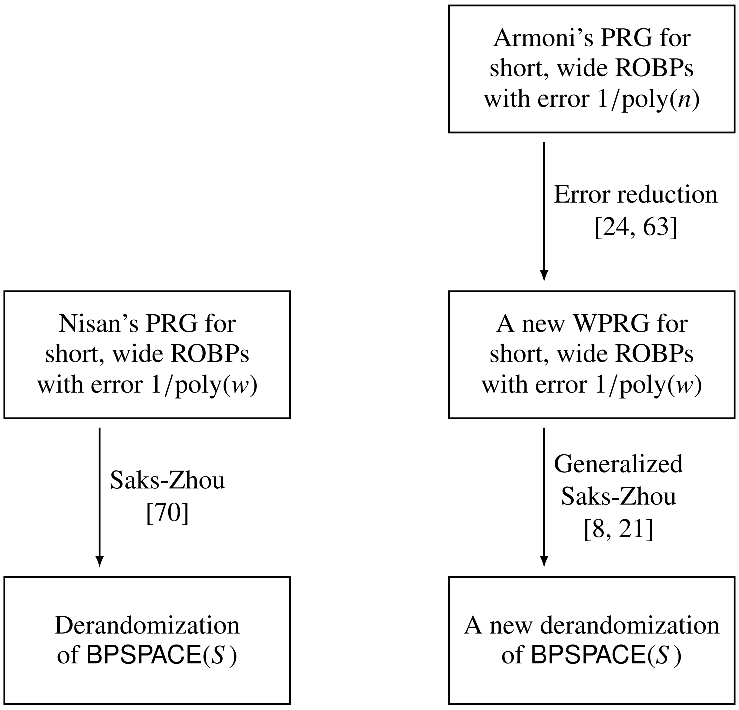

Armoni's PRG for short, wide ROBPs with error $1/\mathrm{poly}(n)$
Error reduction [24, 63]
Nisan's PRG for short, wide ROBPs with error $1/\mathrm{poly}(w)$
A new WPRG for short, wide ROBPs with error $1/\mathrm{poly}(w)$
Saks-Zhou [70]
Generalized Saks-Zhou [8, 21]
Derandomization of $\mathsf{BPSPACE}(S)$
A new derandomization of $\mathsf{BPSPACE}(S)$

Figure 2: Saks and Zhou's derandomization of $\mathsf{BPSPACE}$ [70] (left) vs. the new and improved derandomization of $\mathsf{BPSPACE}$ (Theorem 7, right).

in space $O(\log(wn) \cdot \sqrt{\log n})$. Cohen, Doron, and Sberlo show how to solve the problem in space $\tilde{O}(\log w \cdot \sqrt{\log n} + \log n)$ [25], which is a significant improvement in the regime $n \gg w$. Their algorithm combines the Saks-Zhou algorithm with Richardson iteration, but in a different way than the proof of Theorem 7.

5 Spectral Expander Graphs

Let us consider one more natural problem that is essentially complete for BPL.

- **Input:** A directed graph G, two vertices s and t, and two positive integers k and m (represented in unary).
- **Output:** The probability that a k-step random walk starting at s ends at t, to within an additive error of $1/m$.

An appealing special case is when G is undirected. As mentioned previously, Reingold designed a deterministic log-space algorithm to determine whether there *exists* a path from s to t in an undirected graph G [65], which, intuitively, corresponds to the case $k = \infty$. A recent line of work has studied the case that k is finite, and in particular, k might be smaller than the mixing time of G [53, 54, 1]. For any k, Ahmadinejad, Kelner, Murtagh, Peebles, Sidford, and Vadhan gave an algorithm for computing k-step random walk probabilities in undirected graphs that runs in near-logarithmic space [1].

Theorem 8 (Estimating random-walk probabilities in undirected graphs [1])**.** *Given an undirected graph (or, more generally, an Eulerian digraph) G, two vertices s and t, and positive integers k and m represented in unary, it is possible to deterministically compute the probability that a length-k random walk starting at s arrives at t to within additive error $1/m$ in space $\tilde{O}(\log N)$, where N is the bit-length of the input.*

One of the (many) ideas in the proof of Theorem 8 is to use *expander graphs* to take a certain type of pseudorandom walk through G instead of a truly random walk. There is a long history of using expanders as tools for space-bounded derandomization, going back to work by Ajtai, Komlós, and Szemerédi [2]. Modern work on L vs. BPL continues to develop new ways of using and analyzing expanders – our fourth technical "theme."

5.1 The Derandomized Square

In more detail, the proof of Theorem 8 uses expanders via Rozenman and Vadhan's *derandomized square* operation [68]. For simplicity, consider a D-regular undirected graph G. By definition, a single step in the *square graph* G^2 consists of two steps in the original graph G. Effectively, this means that squaring G places a clique on the D neighbors of each vertex v. The idea of the derandomized square is to instead place an expander graph on the neighbors of v, thereby producing a sparse approximation to G^2.

In Rozenman and Vadhan's original paper, they prove that the *spectral expansion* of the derandomized square is almost as good as that of G^2 [68]. They use this bound to derive an alternative proof that undirected connectivity is in L [68]. Recent work [53, 54, 1] shows that the derandomized square approximates G^2 in much stronger senses. The proof of Theorem 8 combines this analysis with several other techniques, including the inverse Laplacian perspective and error reduction methods.

5.2 The INW Generator

The derandomized square operation also has connections to the PRG approach to L vs. BPL, and in particular to a PRG by Impagliazzo, Nisan, and Wigderson [43] (the "INW generator"). The INW generator samples n pseudorandom bits as follows:

1. Recursively construct a PRG $G\colon \{0,1\}^s \to \{0,1\}^{n/2}$.
2. Sample a uniform random vertex X and a uniform random neighbor Y in a low-degree expander graph on 2^s vertices.
3. Output the concatenation $G(X) \circ G(Y)$.

Several decades after its introduction [43], we are still learning more and more about what the INW generator is capable of. It has been shown to work particularly well for (standard-order) *regular* and *permutation* ROBPs, defined next.

Definition 7 (Regular and permutation ROBPs)**.** Let f be a width-w length-n standard-order ROBP with transition functions $f_1, \dots, f_n\colon [w] \times \{0,1\} \to [w]$. We say that f is a *permutation ROBP* if, for every $i \in [n]$ and every $b \in \{0,1\}$, the function $f_i(\cdot, b)$ is a permutation on $[w]$. More generally, we say that f is *regular* if, for every $i \in [n]$ and every $u \in [w]$, we have $|f_i^{-1}(u)| = 2$.

Regular and permutation ROBPs have been studied extensively over the course of roughly the past decade [16, 17, 27, 46, 75, 66, 19, 1, 40, 62, 63, 26, 64, 11, 35, 48]. We now have various types of pseudorandomness results for regular or permutation ROBPs that are superior to the best corresponding results for general ROBPs, including constructions of PRGs [16, 17, 27, 46, 75, 66, 19, 40, 48], WPRGs [63], and HSGs [16, 11]. In many cases, the proofs consist of improved analyses of the classic INW construction [43] (with modified parameters). In other cases, the INW generator is one of multiple ingredients.

5.3 Unbounded-Width ROBPs

The first few papers on regular and permutation ROBPs [16, 17, 27, 46, 75, 66] focused on constant-width programs. Arguably the most important case is that of polynomial-width programs. The trend recently has been to study the intriguing setting of *unbounded-width* programs [40, 62, 63, 64, 11, 35, 48].

Without further constraints, unbounded-width standard-order permutation ROBPs are too powerful to be interesting: they can compute all Boolean functions. Therefore, we assume that the program has a bounded number of *accepting states* in the final layer. Admittedly, width is a more natural complexity measure than the number of accepting states, but it turns out that programs with a bounded number of accepting states have some interesting properties. Even with just *one* accept state, exponential-width standard-order permutation ROBPs can compute doubly-exponentially many distinct functions:

Proposition 1 ([40])**.** *Let $n \in \mathbb{N}$ be a positive even integer, and let $\pi \colon \{0,1\}^{n/2} \to \{0,1\}^{n/2}$ be a permutation. There exists a width-$(2^{n/2})$ length-n standard-order permutation ROBP f computing the following function:*

$$f(x, y) = 1 \iff \pi(x) = y.$$

(Briefly, to prove Proposition 1, we use the state space $\{0,1\}^{n/2}$. The all-zeroes state is the start state and the unique accepting state. We XOR x into our state, then apply π to our state, then XOR y into our state.) On the other hand, one can check that the majority function on three bits cannot be computed by a standard-order regular ROBP with a single accept vertex, no matter how wide the program is. Thus, these strange unbounded-width models have both dramatic strengths and dramatic weaknesses. One of the most striking results in this area is the following theorem by Pyne and Vadhan [63].

Theorem 9 (WPRGs for unbounded-width permutation ROBPs [63])**.** *For every $n \in \mathbb{N}$ and $\varepsilon > 0$, there is an explicit ε-WPRG for unbounded-width standard-order permutation ROBPs with a single accept state with seed length*

$$\tilde{O}\left(\log^{3/2} n + \log n \cdot \sqrt{\log(1/\varepsilon)} + \log(1/\varepsilon)\right).$$

Theorem 9 has implications for the more conventional setting of bounded-width standard-order permutation ROBPs. Every ε-WPRG for programs with one accepting state automatically (εm)-fools programs with m accepting states. Therefore, Theorem 9 implies an explicit WPRG for width-n length-n standard-order permutation ROBPs (with any number of accepting vertices) with error $1/n$ and seed length $\tilde{O}(\log^{3/2} n)$, compared to Nisan's $O(\log^2 n)$ bound.

Theorem 9 also helps to clarify the importance of weights. When $\varepsilon = 1/n$, the seed length in Theorem 9 is $\tilde{O}(\log^{3/2} n)$. In contrast, Hoza, Pyne, and Vadhan proved that every *unweighted* PRG that $(1/n)$-fools unbounded-width standard-order permutation ROBPs with a single accept vertex must have seed length $\Omega(\log^2 n)$ [40]. Therefore, in at least one natural setting, WPRGs are *intrinsically more powerful* than traditional PRGs.

The proof of Theorem 9 uses the INW generator, the inverse Laplacian perspective, and error reduction techniques, among other ideas.

5.4 The Permutation Case and the Monotone Case: Opposite Extremes

Why study regular and permutation ROBPs? The main reason is the hope that studying these special cases will lead to improvements in the general case. Indeed, there is a reduction showing that good PRGs or HSGs for polynomial-width standard-order *regular* ROBPs imply good PRGs or HSGs for *all* polynomial-width standard-order ROBPs [67, 11].[12]

In addition to that reduction [67, 11], there is another approach for constructing PRGs for constant-width standard-order ROBPs (albeit a vague and speculative one). At an intuitive level, one can argue that permutation ROBPs and monotone ROBPs are "opposites" of one another. In a permutation ROBP, edges with the same label never collide, whereas in a monotone ROBP, the only way that a layer can do any nontrivial computation is by introducing collisions. Now, we have one set of techniques that works well for (standard-order) permutation ROBPs: spectral expanders and the INW generator. Meanwhile, we have another set of techniques that works well for (arbitrary-order) monotone ROBPs: iterated restrictions with early termination. Given that these two sets of techniques cover two "extreme" classes of constant-width ROBPs, it is natural to try to combine the two sets of techniques. Could this approach yield an explicit PRG that fools *all* width-w standard-order ROBPs, with seed length $\tilde{O}(\log n)$ when w is a constant? The idea might sound a bit naïve or fantastical, especially considering the difficulty discussed in Section 2.4. Remarkably, however, Meka, Reingold, and Tal proved that the answer is yes for the case $w = 3$ [50].

Theorem 10 (PRGs for width-3 ROBPs [50])**.** *For every $n \in \mathbb{N}$ and $\varepsilon > 0$, there is an explicit ε-PRG for width-3 standard-order ROBPs with seed length $\tilde{O}(\log n \cdot \log(1/\varepsilon))$.*

[12] Note that Theorem 8 implies a *non-black-box* algorithm for estimating the expectation of a given standard-order regular ROBP in near-logarithmic space. Unfortunately, the reduction from the general case to the regular case does not work in the non-black-box setting.

To prove Theorem 10, Meka, Reingold, and Tal first show how to sample pseudorandom restrictions that preserve the expectation of width-3 arbitrary-order ROBPs. For this first step, one can alternatively use Forbes and Kelley's analysis (Theorem 3), which works more generally for width-w arbitrary-order ROBPs where w is small. (The papers of Forbes and Kelley [31] and Meka, Reingold, and Tal [50] are independent.)

Next, Meka, Reingold, and Tal show that width-3 arbitrary-order ROBPs simplify after a few pseudorandom restrictions [50]. And what does "simplify" mean in this context? Roughly speaking, they show that the program becomes *more and more permutation-like* as the restrictions are applied. After $\mathrm{poly}(\log\log(n/\varepsilon))$ many restrictions, they terminate the restriction process and apply the INW generator [43] as the final step. Building on Braverman, Rao, Raz, and Yehudayoff's analysis [16], they show that the INW generator fools these "highly permutation-like" ROBPs with seed length $\tilde{O}(\log n \cdot \log(1/\varepsilon))$ [50] (in the standard-order case).

It remains an open problem to design an explicit PRG (or WPRG or HSG) for width-4 standard-order ROBPs with seed length $o(\log^2 n)$.

6 Conclusions

We continue to make steady, substantial progress toward proving $\mathsf{L} = \mathsf{BPL}$. The past few years alone have yielded many exciting results and developments. The problem remains challenging, but there does not seem to be any firm obstacle preventing further breakthroughs.

7 Acknowledgments

I thank Paul Beame, Lijie Chen, Oded Goldreich, and Edward Pyne for helpful comments on drafts of this article. Part of this work was done while I was visiting the Simons Institute for the Theory of Computing. This article is based on a presentation that I first gave at Oberwolfach Workshop 2146 on Complexity Theory.

References

[1] AmirMahdi Ahmadinejad, Jonathan Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil Vadhan. High-precision estimation of random walks in small space. In *Proceedings of the 61st Symposium on Foundations of Computer Science*, pages 1295–1306, 2020.

[2] Miklós Ajtai, János Komlós, and Endre Szemerédi. Deterministic simulation in logspace. In *Proceedings of the 19th Symposium on Theory of Computing (STOC)*, pages 132–140, 1987.

[3] Miklós Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant-depth circuits. *Advances in Computing Research – Randomness and Computation*, 5:199–23, 1989.

[4] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, László Lovász, and Charles Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. In *Proceedings of the 20th Symposium on Foundations of Computer Science (FOCS)*, pages 218–223, 1979.

[5] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures Algorithms*, 3(3):289–304, 1992.

[6] Alexander E. Andreev, Andrea E. F. Clementi, and José D. P. Rolim. A new general derandomization method. *J. ACM*, 45(1):179–213, 1998.

[7] Alexander E. Andreev, Andrea E. F. Clementi, José D. P. Rolim, and Luca Trevisan. Weak random sources, hitting sets, and BPP simulations. *SIAM J. Comput.*, 28(6):2103–2116, 1999.

[8] Roy Armoni. On the derandomization of space-bounded computations. In *Proceedings of the 2nd International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, pages 47–59, 1998.

[9] Roy Armoni, Amnon Ta-Shma, Avi Wigderson, and Shiyu Zhou. An $O(\log(n)^{4/3})$ space algorithm for (s, t) connectivity in undirected graphs. *J. ACM*, 47(2):294–311, 2000.

[10] Greg Barnes and Walter L. Ruzzo. Undirected s-t connectivity in polynomial time and sublinear space. *Comput. Complexity*, 6(1):1–28, 1997.

[11] Andrej Bogdanov, William M. Hoza, Gautam Prakriya, and Edward Pyne. Hitting Sets for Regular Branching Programs. In *Proceedings of the 37th Computational Complexity Conference (CCC 2022)*, pages 3:1–3:22, 2022.

[12] Andrej Bogdanov, Periklis A. Papakonstantinou, and Andrew Wan. Pseudorandomness for read-once formulas. In *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS)*, pages 240–246, 2011.

[13] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Inform. and Control*, 58(1-3):113–136, 1983.

[14] Allan Borodin, Stephen A. Cook, Patrick W. Dymond, Walter L. Ruzzo, and Martin Tompa. Two applications of inductive counting for complementation problems. *SIAM J. Comput.*, 18(3):559–578, 1989.

[15] Mark Braverman, Gil Cohen, and Sumegha Garg. Pseudorandom pseudo-distributions with near-optimal error for read-once branching programs. *SIAM Journal on Computing*, 49(5):STOC18–242–STOC18–299, 2020.

[16] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014.

[17] Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *Proceedings of the 51st Symposium on Foundations of Computer Science (FOCS)*, pages 30–39, 2010.

[18] Harry Buhrman and Lance Fortnow. One-sided versus two-sided error in probabilistic computation. In *Proceedings of the 16th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 100–109, 1999.

[19] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. *Theory Comput.*, 15:Paper No. 10, 2019.

[20] Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *Proceedings of the 50th Symposium on Theory of Computing (STOC)*, pages 363–375, 2018.

[21] Eshan Chattopadhyay and Jyun-Jie Liao. Optimal Error Pseudodistributions for Read-Once Branching Programs. In *Proceedings of the 35th Computational Complexity Conference (CCC)*, pages 25:1–25:27, 2020.

[22] Sitan Chen, Thomas Steinke, and Salil Vadhan. Pseudorandomness for read-once, constant-depth circuits. arXiv preprint 1504.04675, 2015.

[23] Kuan Cheng and William M. Hoza. Hitting Sets Give Two-Sided Derandomization of Small Space. In *Proceedings of the 35th Computational Complexity Conference (CCC)*, pages 10:1–10:25, 2020.

[24] Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted PRGs against read once branching programs. In *Proceedings of the 36th Computational Complexity Conference*, pages 22:1–22:17, 2021.

[25] Gil Cohen, Dean Doron, and Ori Sberlo. Approximating large powers of stochastic matrices in small space. ECCC preprint TR22-008, 2022.

[26] Gil Cohen, Noam Peri, and Amnon Ta-Shma. Expander random walks: a Fourier-analytic approach. In *Proceedings of the 53rd Annual Symposium on Theory of Computing (STOC)*, pages 1643–1655, 2021.

[27] Anindya De. Pseudorandomness for permutation and regular branching programs. In *Proceedings of the 26th Conference on Computational Complexity (CCC)*, pages 221–231, 2011.

[28] Dean Doron, Pooya Hatami, and William M. Hoza. Near-optimal pseudorandom generators for constant-depth read-once formulas. In *Proceedings of the 34th Computational Complexity Conference (CCC)*, pages 16:1–16:34, 2019.

[29] Dean Doron, Pooya Hatami, and William M. Hoza. Log-Seed Pseudorandom Generators via Iterated Restrictions. In *Proceedings of the 35th Computational Complexity Conference (CCC)*, pages 6:1–6:36, 2020.

[30] Dean Doron, Raghu Meka, Omer Reingold, Avishay Tal, and Salil Vadhan. Pseudorandom generators for read-once monotone branching programs. In *Proceedings of the 25th International Conference on Randomization and Computation (RANDOM)*, pages 58:1–58:21, 2021.

[31] Michael A. Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *Proceedings of the 59th Symposium on Foundations of Computer Science (FOCS)*, pages 946–955, 2018.

[32] Dmitry Gavinsky, Shachar Lovett, and Srikanth Srinivasan. Pseudorandom generators for read-once ACC^0. In *Proceedings of the 27th Conference on Computational Complexity (CCC)*, pages 287–297, 2012.

[33] John Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6(4):675–695, 1977.

[34] Oded Goldreich, Salil Vadhan, and Avi Wigderson. Simplified derandomization of BPP using a hitting set generator. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, volume 6650 of *Lecture Notes in Comput. Sci.*, pages 59–67. Springer, Heidelberg, 2011.

[35] Louis Golowich and Salil Vadhan. Pseudorandomness of Expander Random Walks for Symmetric Functions and Permutation Branching Programs. In *Proceedings of the 37th Computational Complexity Conference (CCC)*, pages 27:1–27:13, 2022.

[36] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd Symposium on Foundations of Computer Science (FOCS)*, pages 120–129, 2012.

[37] Elad Haramaty, Chin Ho Lee, and Emanuele Viola. Bounded independence plus noise fools products. *SIAM J. Comput.*, 47(2):493–523, 2018.

[38] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. Fooling constant-depth threshold circuits. In *Proceedings of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 104–115, 2022 (albeit "FOCS 2021").

[39] William M. Hoza. Better pseudodistributions and derandomization for space-bounded computation. In *Proceedings of the 25th International Conference on Randomization and Computation (RANDOM)*, pages 28:1–28:23, 2021.

[40] William M. Hoza, Edward Pyne, and Salil Vadhan. Pseudorandom Generators for Unbounded-Width Permutation Branching Programs. In *Proceedings of the 12th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 7:1–7:20, 2021.

[41] William M. Hoza and David Zuckerman. Simple optimal hitting sets for small-success **RL**. *SIAM J. Comput.*, 49(4):811–820, 2020.

[42] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. *J. ACM*, 66(2):Art. 11, 16, 2019.

[43] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th Symposium on Theory of Computing (STOC)*, pages 356–364, 1994.

[44] A. Joffe. On a set of almost deterministic k-independent random variables. *Ann. Probability*, 2(1):161–162, 1974.

[45] H. Jung. Relationships between probabilistic and deterministic tape complexity. In *Proceedings of the 10th Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 339–346, 1981.

[46] Michal Koucký, Prajakta Nimbhorkar, and Pavel Pudlák. Pseudorandom generators for group products. In *Proceedings of the 43rd Symposium on Theory of Computing (STOC)*, pages 263–272, 2011.

[47] Chin Ho Lee. Fourier bounds and pseudorandom generators for product tests. In *Proceedings of the 34th Computational Complexity Conference (CCC)*, pages 7:1–7:25, 2019.

[48] Chin Ho Lee, Edward Pyne, and Salil Vadhan. Fourier growth of regular branching programs. ECCC preprint TR22-034, 2022.

[49] Chin Ho Lee and Emanuele Viola. More on bounded independence plus noise: pseudorandom generators for read-once polynomials. *Theory Comput.*, 16:Paper No. 7, 50, 2020.

[50] Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom generators for width-3 branching programs. In *Proceedings of the 51st Symposium on Theory of Computing (STOC)*, pages 626–637, 2019.

[51] Raghu Meka and David Zuckerman. Pseudorandom generators for polynomial threshold functions. *SIAM J. Comput.*, 42(3):1275–1301, 2013.

[52] Pascal Michel. A survey of space complexity. *Theoret. Comput. Sci.*, 101(1):99–132, 1992.

[53] Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil Vadhan. Derandomization beyond connectivity: undirected Laplacian systems in nearly logarithmic space. *SIAM J. Comput.*, 50(6):1892–1922, 2021.

[54] Jack Murtagh, Omer Reingold, Aaron Sidford, and Salil Vadhan. Deterministic approximation of random walks in small space. *Theory Comput.*, 17:Paper No. 4, 35, 2021.

[55] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.

[56] È. I. Nečiporuk. On a Boolean function. *Dokl. Akad. Nauk SSSR*, 169:765–766, 1966.

[57] N. Nisan, E. Szemeredi, and A. Wigderson. Undirected connectivity in $o(\log^{1.5} n)$ space. In *Proceedings of the 33rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 24–29, 1992.

[58] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[59] Noam Nisan and Amnon Ta-Shma. Symmetric Logspace is closed under complement. *Chicago J. Theoret. Comput. Sci.*, pages Article 1, approx. 11pp., 1995.

[60] Noam Nisan and Avi Wigderson. Hardness vs. randomness. *J. Comput. System Sci.*, 49(2):149–167, 1994.

[61] Noam Nisan and David Zuckerman. Randomness is linear in space. *J. Comput. System Sci.*, 52(1):43–52, 1996.

[62] Edward Pyne and Salil Vadhan. Limitations of the Impagliazzo-Nisan-Wigderson pseudorandom generator against permutation branching programs. In *Proceedings of the 27th International Computing and Combinatorics Conference (COCOON)*, pages 3–12, 2021.

[63] Edward Pyne and Salil Vadhan. Pseudodistributions that beat all pseudorandom generators (extended abstract). In *Proceedings of the 36th Computational Complexity Conference (CCC)*, pages 33:1–33:15, 2021.

[64] Edward Pyne and Salil Vadhan. Deterministic approximation of random walks via queries in graphs of unbounded size. In *Proceedings of the 5th Symposium on Simplicity in Algorithms (SOSA)*, pages 57–67, 2022.

[65] Omer Reingold. Undirected connectivity in log-space. *J. ACM*, 55(4):Art. 17, 24, 2008.

[66] Omer Reingold, Thomas Steinke, and Salil Vadhan. Pseudorandomness for regular branching programs via Fourier analysis. In *Proceedings of the 17th International Workshop on Randomization and Computation (RANDOM)*, pages 655–670, 2013.

[67] Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom walks on regular digraphs and the **RL** vs. **L** problem. In *Proceedings of the 38th Symposium on Theory of Computing (STOC)*, pages 457–466, 2006.

[68] Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pages 436–447, 2005.

[69] Michael Saks. Randomization and derandomization in space-bounded computation. In *Proceedings of the 11th Conference on Computational Complexity (CCC)*, pages 128–149, 1996.

[70] Michael Saks and Shiyu Zhou. $\mathrm{BP_H SPACE}(S) \subseteq \mathrm{DSPACE}(S^{3/2})$. *J. Comput. System Sci.*, 58(2):376–403, 1999.

[71] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. System Sci.*, 4:177–192, 1970.

[72] Janos Simon. On the difference between one and many (preliminary version). In *Proceedings of the 4th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 480–491, 1977.

[73] Janos Simon. On tape-bounded probabilistic Turing machine acceptors. *Theoret. Comput. Sci.*, 16(1):75–91, 1981.

[74] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 81–90. ACM, New York, 2004.

[75] Thomas Steinke. Pseudorandomness for permutation branching programs without the group theory. ECCC preprint TR12-083, 2012.

[76] Thomas Steinke, Salil Vadhan, and Andrew Wan. Pseudorandomness and Fourier-growth bounds for width-3 branching programs. *Theory Comput.*, 13:Paper No. 12, 2017.

[77] Vladimir Trifonov. An $O(\log n \log\log n)$ space algorithm for undirected *st*-connectivity. *SIAM J. Comput.*, 38(2):449–483, 2008.

[78] Yoav Tzur. Notions of weak pseudorandomness and $GF(2^n)$-polynomials. M.Sc. thesis, Weizmann Institute of Science, 2009.

The Logic in Computer Science Column

by

Yuri Gurevich

Computer Science & Engineering
University of Michigan, Ann Arbor, Michigan, USA
gurevich@umich.edu

A Surprising Relationship Between Descriptive Complexity and Proof Complexity

Yijia Chen
Shanghai Jiao Tong University
Department of Computer Science
`yijia.chen@cs.sjtu.edu.cn`

Jörg Flum
Albert-Ludwigs-Universität Freiburg i. Br
Mathematisches Institut
`joerg.flum@math.uni-freiburg.de`

Moritz Müller
Universität Passau
Fakultät für Informatik und Mathematik
`moritz.mueller@uni-passau.de`

In [4] Gurevich conjectured that there is no logic that captures PTIME. This is the main open problem of descriptive complexity. A central issue in proof complexity is whether p-optimal proof systems for the set TAUT of tautologies of propositional logic exist. It appears explicitly in [5] and implicitly already in the foundational paper of Cook and Reckow [3].

Around ten years ago Chen and Flum [1] showed that these two problems are tightly related: there is a p-optimal proof system for TAUT if and only if a certain logic considered by Gurevich in [4] captures PTIME. How surprising is this equivalence? It turns out that both statements are equivalent to the membership of a parameterized halting problem for Turing machines in a certain complexity class of parameterized complexity theory [2].

The purpose of this note is to present a short direct proof of (a variant of) the mentioned equivalence. It is intended to be accessible to non-experts. We follow the established style of this column and present the proof in dialogue form.

Professor Gurevich **G** *talks to one of his students* **SG**.

SG: You conjecture that there is no logic capturing PTIME. What is the underlying notion of a logic here?

G: Roughly speaking a *logic* is given by a map L and a binary relation $\models_L$. The map L assigns to every vocabulary τ, i.e., to every finite set of relation symbols, a set $L[\tau]$ of strings, the so-called *τ-sentences of L*. If $\mathcal{A} \models_L \varphi$, then $\mathcal{A}$ is a finite τ-structure and φ a τ-sentence of L. The map and the relation have to satisfy some natural properties, which I will explain if necessary. A sentence $\varphi \in L[\tau]$ *defines* the class $\mathrm{Mod}_L(\varphi)$ of τ-structures $\mathcal{A}$ such that $\mathcal{A} \models_L \varphi$.

SG: What does it mean that a logic captures PTIME?

G: For some vocabulary τ we view all instances of a given *(computational) problem* as τ-structures. We identify the problem with the class of its YES-instances. A logic *captures* PTIME if and only if its sentences define precisely the (classes of YES-instances of) problems in PTIME. Again some additional properties are required, which I will explain if necessary.

SG: Consider the logic L_1 where, for any τ, $L_1[\tau]$ is the set of polynomial time *clocked* algorithms (i.e., each algorithm comes with an explicit polynomial time bound). For such an algorithm $\mathbb{A}$ and any τ-structure declare $\mathcal{A} \models_{L_1} \mathbb{A}$ to mean that $\mathbb{A}$ accepts $\mathcal{A}$. Why does this logic not capture PTIME?

G: Because of an additional property required of a logic: $\mathrm{Mod}_L(\varphi)$ has to be closed under isomorphism for any sentence φ. A polynomial time algorithm might reject (the binary encoding of) some structure but accept (the binary encoding of) an isomorphic one.

SG: OK, then consider the logic L_2 where $L_2[\tau]$ for any τ is the set of polynomial time algorithms $\mathbb{A}$ that are *invariant*: if $\mathcal{A}$ and $\mathcal{B}$ are isomorphic τ-structures, then $\mathbb{A}$ accepts $\mathcal{A}$ if and only if $\mathbb{A}$ accepts $\mathcal{B}$. Define $\models_{L_2}$ as for L_1. Why does this logic not capture PTIME?

G: Because of a further additional property required of a logic: for every τ, the set $L[\tau]$ has to be decidable. But it is undecidable whether a clocked polynomial time algorithm is invariant.

SG: A further attempt with the logic L_3. Let $L_3[\tau] := L_1[\tau]$ but now define $\mathcal{A} \models_{L_3} \mathbb{A}$ to mean that $\mathbb{A}$ accepts $\mathcal{A}$ and $\mathbb{A}$ is invariant. For non-invariant $\mathbb{A}$ we have $\mathrm{Mod}_{L_3}(\mathbb{A}) = \emptyset$. Why does this logic not capture PTIME?

G: Because of an extra condition in what it means that a logic L captures PTIME: we require that for each τ there exists a *model-checker*, i.e., an algorithm that, given a τ-structure $\mathcal{A}$ and $\varphi \in L[\tau]$, decides whether $\mathcal{A} \models_L \varphi$. Such an algorithm does not exist for L_3.

SG: My last attempt. Set $L_4[\tau] := L_1[\tau]$ and define $\mathcal{A} \models_{L_4} \mathbb{A}$ to mean that $\mathbb{A}$ accepts $\mathcal{A}$ and $\mathbb{A}$ is n-invariant where n is the size of the universe of $\mathcal{A}$. That $\mathbb{A}$ is

n-invariant means: if $\mathcal{A}$ and $\mathcal{B}$ are isomorphic τ-structures of size at most n, then $\mathbb{A}$ accepts $\mathcal{A}$ if and only if $\mathbb{A}$ accepts $\mathcal{B}$. Why does this logic not capture PTIME?

G: Well, we do not know whether L_4 captures PTIME. More precisely, we do not know whether L_4 satisfies a final condition on a logic L capturing PTIME. This condition requires that for each fixed φ, the model-checker runs in polynomial time when restricted to inputs $(\mathcal{A}, \varphi)$.

SG: Where does this requirement come from?

G: Intuitively, it allows to view a logic capturing PTIME as a high level programming language for PTIME. More precisely, every φ in the logic is viewed as a program, and the model-checker is an interpreter which executes this program on any structure $\mathcal{A}$ in time polynomial in the size of $\mathcal{A}$. In addition, for every PTIME problem we can write such a program φ.

SG: To sum up, a logic capturing PTIME consists of a map L from vocabularies τ to sets $L[\tau]$ of τ-sentences, a relation $\models_L$ between τ-structures and τ-sentences, and a *model-checker*, an algorithm that given a τ-structure $\mathcal{A}$ and $\varphi \in L[\tau]$ decides whether $\mathcal{A} \models_L \varphi$, such that for every τ

- the set $L[\tau]$ is decidable;
- for every $\varphi \in L[\tau]$, the class $\mathrm{Mod}_L(\varphi) = \{\mathcal{A} \mid \mathcal{A} \models_L \varphi\}$ is closed under isomorphism;
- every problem in PTIME, if viewed as a class of τ-structures, equals $\mathrm{Mod}_L(\varphi)$ for some $\varphi \in L[\tau]$;
- for every fixed $\varphi \in L[\tau]$, the runtime of the model-checker on $(\mathcal{A}, \varphi)$ is polynomial in the size of $\mathcal{A}$.

G: Yes, this is how the question whether there exists a logic capturing PTIME is formulated in [4]. Your *naive logic* L_4 satisfies the first three items but I conjecture it does not satisfy the last one.

SG: It would be sufficient to have an algorithm that given $(\mathbb{A}, n)$ decides whether $\mathbb{A}$ is n-invariant in time $p_{\mathbb{A}}(n)$ where $p_{\mathbb{A}}$ is a polynomial that may depend on $\mathbb{A}$.

G: In fact, this is also necessary for L_4 capturing PTIME: let $\neg\mathbb{A}$ behave as $\mathbb{A}$ but flip the answer, i.e., $\neg\mathbb{A}$ accepts if and only if $\mathbb{A}$ rejects. To decide whether $\mathbb{A}$ is n-invariant, take some arbitrary structure $\mathcal{A}$ of size n and use the model-checker to check whether $\mathcal{A} \models_{L_4} \mathbb{A}$ or $\mathcal{A} \models_{L_4} \neg\mathbb{A}$.

Professor Cook **C** *talks to one of his students* **SC**.

SC: Some conjecture that there is no p-optimal proof system for the set TAUT of tautologies of propositional logic. What does this mean, and, in particular, what is the underlying notion of a proof system here?

C: A *proof system* is a polynomial time computable function P from the set of binary strings onto TAUT. A binary string x is a *P-proof* of the tautology $P(x)$. Being *p-optimal* means that for every other proof system P' there is a polynomial time computable function T translating P'-proofs into P-proofs of the same tautologies, i.e., such that $P'(x) = P(T(x))$ for all binary strings x.

SC: Define the proof system P_0 as follows. On input $(P, x, 1^t)$ where P is (an algorithm computing a) a proof system, x is a binary string, and $t \in \mathbb{N}$, simulate P on x for at most t many steps; if the simulation halts, return its output $P(x)$; otherwise, return some fixed tautology, say $(X \vee \neg X)$; also return $(X \vee \neg X)$ on inputs that are not of the required form. This is a map onto TAUT. If P is a proof system and p is a polynomial bound for its running time, then $x \mapsto (P, x, 1^{p(|x|)})$ is a translation as required. Why isn't P_0 a p-optimal proof system?

C: Because it is not decidable whether a given polynomial time algorithm is a proof system.

SC: Well, then we define P_1 as P_0 but now we consider inputs $(\mathbb{A}, x, 1^t)$ where $\mathbb{A}$ is an arbitrary (clocked) polynomial time algorithm. On such an input, P_1 first spends t steps to check whether $\mathbb{A}$ is $|x|$-sound before simulating it and proceeding as P_0. That $\mathbb{A}$ is *n-sound* means: $\mathbb{A}(y)$ is a tautology for all y with $|y| \leqslant n$. If the check fails or P_1 runs out of time, then it outputs $(X \vee \neg X)$.

C: It is unknown whether your *naive proof system* P_1 is p-optimal. Your translation $x \mapsto (\mathbb{A}, x, 1^{p(|x|)})$ needs a polynomial $p(|x|)$ so that the simulation and the $|x|$-soundness check can be done in time $p(|x|)$.

SC: It would be sufficient to have an algorithm that given $(\mathbb{A}, n)$ decides whether $\mathbb{A}$ is n-sound in time $p_{\mathbb{A}}(n)$ for some polynomial $p_{\mathbb{A}}$ that may depend on $\mathbb{A}$.

The two students **SG** *and* **SC** *meet,* **SC** *asks* **SG** *for her interests, and* **SG** *recounts her conversation with* **G**.

SC: After listening to you I believe that the existence of a p-optimal proof system implies that the naive logic L_4 is a logic for PTIME.

SG: Why?

SC: Assume there is a p-optimal proof system P. By a classical result of Levin [6], P has an optimal inverter $\mathbb{I}$. Being an *inverter* means that $\mathbb{I}$, given a tautology, outputs a P-proof of it; on other inputs $\mathbb{I}$ diverges. Being *optimal* means: for every inverter $\mathbb{I}'$ there is a polynomial p' such that $t_{\mathbb{I}}(x) \leqslant p'(t_{\mathbb{I}'}(x) + |x|)$ for every tautology x. Here, $t_{\mathbb{I}}(x)$ and $t_{\mathbb{I}'}(x)$ denote the runtimes of $\mathbb{I}$ and $\mathbb{I}'$ on x.

For $(\mathbb{A}, n)$, where $\mathbb{A}$ is a polynomial time algorithm and $n \geqslant 1$, to decide whether $\mathbb{A}$ is n-invariant is a problem in coNP (the complement is in NP!). By coNP-completeness of TAUT, there is a polynomial time function assigning to $(\mathbb{A}, n)$ a propositional formula $F^n_{\mathbb{A}}$ such that $F^n_{\mathbb{A}}$ is a tautology if and only if $\mathbb{A}$ is n-invariant.

Let $\mathbb{A}$ be invariant. Then all $F^n_{\mathbb{A}}$ are tautologies. One easily defines a proof system P' that has 1^n as a P'-proof of $F^n_{\mathbb{A}}$. By p-optimality of the proof system P there is a translation T, i.e., $P' = P \circ T$. Define an inverter $\mathbb{I}'$ of P that maps $F^n_{\mathbb{A}}$ to $T(1^n)$ – we can assume that one can recover n from $F^n_{\mathbb{A}}$ in polynomial time. By optimality of the inverter $\mathbb{I}$ of P, also $\mathbb{I}$ on $F^n_{\mathbb{A}}$ needs time $p_{\mathbb{A}}(n)$ for some polynomial $p_{\mathbb{A}}$.

SG: But runtime $p_{\mathbb{A}}(n)$ is ensured only on inputs $(\mathbb{A}, n)$ where $\mathbb{A}$ is invariant. On other inputs ($\mathbb{I}$ and) your algorithm might even diverge.

SC: Right. So run the algorithm in parallel with some brute force procedure that on $(\mathbb{A}, n)$ computes the minimal m such that $\mathbb{A}$ is not m-invariant; for invariant $\mathbb{A}$ this procedure does not halt. Otherwise it halts in some time depending only on $\mathbb{A}$. If it halts, check $n < m$. Then the runtime on inputs $(\mathbb{A}, n)$ with non-invariant $\mathbb{A}$ is also bounded as desired.

SG: I'm impressed. Now we know: if p-optimal proof systems exist, then my naive logic captures PTIME.

SG *asks* **SC** *for her interests, and* **SC** *recounts her conversation with* **C**.

SG: After listening to you I believe that if my naive logic captures PTIME, then your naive proof system is p-optimal.

SC: Why?

SG: So far we used formulations like "an algorithm accepts a structure." But what does it mean that an abstract structure is an input to an algorithm? Now we have to be more precise. We use binary codes of structures. For this purpose we assume that we deal with *standard structures*, the universe of a standard structure is the set $[n]$ $(:= \{1, 2, \dots, n\})$ for some $n \geqslant 1$. Of course, every abstract structure is isomorphic to a standard structure. Then, once we have fixed an ordering on the relations of a vocabulary τ, every standard τ-structure corresponds to a unique binary string. To apply the algorithm to the structure means that this string is the input to the algorithm.

So let's come back to our problem. Let $\tau := \{<, \mathit{One}, \mathit{Zero}\}$ with binary $<$ and unary *One* and *Zero*. For a binary string $x = x_1 \dots x_{|x|}$ and a natural number $m > |x|$ let the τ-structure $\mathcal{A}(x, m)$ have universe $[2m]$, interpret $<$ by the natural order on $[2m]$, *One* by $\{i \mid x_i = 1\}$, and *Zero* by $\{i \mid x_i = 0\}$. Given a (standard) τ-structure $\mathcal{B}$, one can check in polynomial time whether $\mathcal{B}$ is isomorphic to some $\mathcal{A}(x, m)$, and in the positive case compute the unique such pair (x, m). Such a $\mathcal{B}$ *codes* an assignment $\alpha_{\mathcal{B}}$ to m propositional variables: assign true or false to the j-th variable depending on whether $2j - 1$ is smaller than $2j$ in the order $<^{\mathcal{B}}$, the interpretation of $<$ in $\mathcal{B}$. Clearly, every assignment to m variables is coded by some $\mathcal{B} \cong \mathcal{A}(x, m)$.

SC: What does this help to reduce my soundness problem to your invariance problem?

SG: Let $\mathbb{A}$ be a polynomial time algorithm and without loss of generality assume that on any input it always outputs a propositional formula. Furthermore let $q_{\mathbb{A}}$ be a strictly increasing polynomial such that $q_{\mathbb{A}}(n)$ is an upper bound for the number of variables of $\mathbb{A}(x)$ for all x with $|x| \leqslant n$.

Define $\mathbb{A}^*$ to check, given a standard τ-structure $\mathcal{B}$, whether it is isomorphic to some $\mathcal{A}(x, q_{\mathbb{A}}(|x|) + 1)$. If not $\mathbb{A}^*$ rejects $\mathcal{B}$; otherwise, it computes $\mathbb{A}(x)$ which we have assumed to be a propositional formula. Let X be the "first" variable not occurring in $\mathbb{A}(x)$. Then the formula $(\mathbb{A}(x) \vee X)$ is satisfiable and $((\mathbb{A}(x) \vee X)$ is a tautology if and only if $\mathbb{A}(x)$ is a tautology). The algorithm $\mathbb{A}^*$ accepts $\mathcal{B}$ if $\alpha_{\mathcal{B}}$ satisfies $(\mathbb{A}(x) \vee X)$ and otherwise rejects $\mathcal{B}$.

Then $\mathbb{A}$ is n-sound if and only if $\mathbb{A}^*$ is $2(q_{\mathbb{A}}(n) + 1)$-invariant.

SC: Why?

SG: If $\mathbb{A}$ is not n-sound, some $\mathbb{A}(x)$ with $|x| \leqslant n$ is not a tautology. Then $(\mathbb{A}(x) \vee X)$ is not a tautology but satisfiable. Choose a satisfying and a falsifying assignment for $(\mathbb{A}(x) \vee X)$. There are two structures $\mathcal{B}_1$ and $\mathcal{B}_2$ isomorphic to $\mathcal{A}(x, q_{\mathbb{A}}(|x|) + 1)$ such that $\alpha_{\mathcal{B}_1}$ and $\alpha_{\mathcal{B}_2}$ are these assignments. Then $\mathbb{A}^*$ accepts $\mathcal{B}_1$ but rejects $\mathcal{B}_2$. Thus $\mathbb{A}^*$ is not $2(q_{\mathbb{A}}(|x|) + 1)$-invariant and hence not $2(q_{\mathbb{A}}(n) + 1)$-invariant.

Now assume that $\mathbb{A}$ is n-sound. Assume $\mathbb{A}^*$ accepts a size $\leqslant 2(q_{\mathbb{A}}(n) + 1)$ structure $\mathcal{B}$. Then $\mathcal{B} \cong \mathcal{A}(x, q_{\mathbb{A}}(|x|)+1)$ for some $|x| \leqslant n$. Conversely, every such $\mathcal{B}$ has size at most $2(q_{\mathbb{A}}(n) + 1)$ and is accepted by $\mathbb{A}^*$ if $\alpha_{\mathcal{B}}$ satisfies $(\mathbb{A}(x) \vee X)$. By n-soundness of $\mathbb{A}$, the formula $(\mathbb{A}(x) \vee X)$ is a tautology and in particular, satisfied by $\alpha_{\mathcal{B}}$.

SC: To sum up, we proved:

Theorem. *Statements (2), (3) and (4) are equivalent and they imply (1).*

(1) There is a logic capturing PTIME.

(2) The naive logic captures PTIME.

(3) The naive proof system is p-optimal.

(4) There is a p-optimal proof system.

The students ask around whether (1) is equivalent to some natural weakening of (4) but nobody seems to know anything.

Acknowledgement We thank Albert Atserias and anonymous reviewers for comments on an earlier version of this text.

References

[1] Yijia Chen and Jörg Flum. From almost optimal algorithms to logics for complexity classes via listings and a halting problem. *Journal of the ACM*, 59(4):17:1–17:34, 2012.

[2] Yijia Chen and Jörg Flum. A parameterized halting problem. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, volume 7370 of *Lecture Notes in Computer Science*, pages 364–397. Springer, 2012.

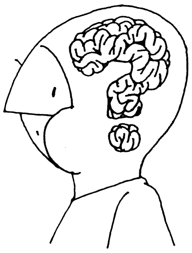
[3] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, 1979.

[4] Yuri Gurevich. Logic and the challenge of computer science. In Egon Börger, editor, *Current Trends in Theoretical Computer Science*, pages 1–57. Computer Science Press, 1988.

[5] Jan Krajíček and Pavel Pudlák. Propositional proof systems, the consistency of first order theories and the complexity of computations. *Journal of Symbolic Logic*, 54(3):1063–1079, 1989.

[6] Leonid A. Levin. Universal sequential search problems. *Problems of Information Transmission*, 9(3):265–266, 1973.

News and Conference Reports



CONCUR THROUGH TIME

Luca Aceto
ICE-TCS, Department of Computer Science,
Reykjavik University
Gran Sasso Science Institute, L'Aquila
luca@ru.is, luca.aceto@gssi.it

Pierluigi Crescenzi
Gran Sasso Science Institute, L'Aquila
pierluigi.crescenzi@gssi.it

1 Introduction

The 33rd edition of the International Conference on Concurrency Theory (CONCUR) will be held in Warsaw, Poland, in the period 16–19 September 2022. The first CONCUR conference dates back to 1990 and was one of the conferences organized as part of the two-year ESPRIT Basic Research Action 3006 with the same name. The CONCUR community has run the conference ever since and established the IFIP WG 1.8 "Concurrency Theory" in 2005 under Technical Committee TC1 Foundations of Computer Science of IFIP[1].

In light of the well-established nature of the CONCUR conference, and spurred by a data- and graph-mining comparative analysis carried out by the second author to celebrate the 50th anniversary of ICALP[2], we undertook a similar study for the CONCUR conference using some, by now classic, tools from network science. Our goal was to try and understand the evolution of the CONCUR conference throughout its history, the ebb and flow in the popularity of some research areas in concurrency theory, and the centrality of CONCUR authors, as measured by several metrics from network science, amongst other topics.

This article reports on our findings. We hope that members of the CONCUR community will enjoy reading it and playing with the web-based resources that accompany this piece. It goes without saying that the data analysis we present has to be taken with a huge pinch of salt and is only meant to provide an overview of the evolution of CONCUR and to be food for thought for the concurrency theory community.

The paper is organized as follows. Section 2 describes the data collection and mining software used for the analysis presented in our study. Section 3 details the evolution of the number of CONCUR papers and authors per year, and Section 4 reports on our findings related to the representation of female authors at the conference. We present data on the evolution of popular research topics in papers presented at CONCUR in Section 5 by analyzing the words

[1] See https://pure.tue.nl/ws/portalfiles/portal/4345371/589768.pdf and https://concurrency-theory.org/organizations/ifip for information on the ESPRIT project and the IFIP "Concurrency Theory" working group, respectively.

[2] See the presentation available at https://slides.com/piluc/icalp-50?token=fl3BBJ8j.

appearing in the paper titles. Section 6 is devoted to a study of the CONCUR collaboration graph. We conclude the article by applying several centrality measures from network science to identify the "most central figures" in the CONCUR community (Section 7).

2 Data collection and mining software

The data collection software has been developed in Java, mostly because this allowed us to take advantage of the Java library available on the DBLP web site[3]. (All the generated graphs are based on the DBLP XML file dated March 1, 2022, and up to the 2021 edition of CONCUR[4].) Note that, even if the first CONCUR took place in August 1990, the collected data include also the papers published in three events devoted to concurrency that took place in July 1984, October 1988, and September 1989, respectively[5]. The basic data mining software has been developed in Julia. Both the Java code and the Julia code are publicly available at the following GitHub repository: `https://github.com/piluc/ConferenceMining`.

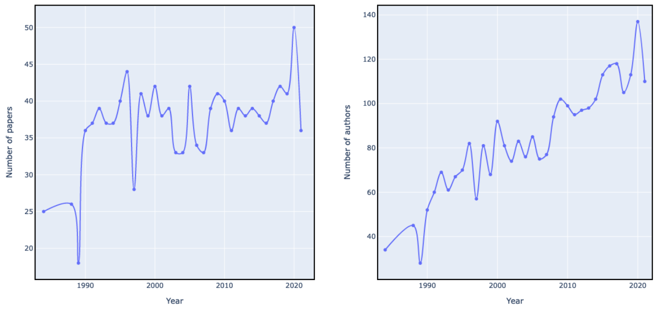


Figure 1: The evolution of the number of CONCUR papers (left) and the number of authors per year (right).

3 Evolution of paper and author numbers

The evolution of the number of CONCUR papers per year is shown in the left part of Figure 1, while the evolution of the number of authors per year is depicted in the right part of that figure. We observe that, while the number of papers per year has been rather stable (approximately

[3] See `https://dblp.org/faq/1474681.html`.

[4] The tables of contents of all the editions of CONCUR are available on the DBLP web site, starting from `https://dblp.org/db/conf/concur/index.html`. The structure of the DBLP XML file, instead, is described in M. Ley, "DBLP – Some Lessons Learned", *Proc. VLDB Endow.*, 2(2): 1493-1500 (2009).

[5] These three events, which predate the first CONCUR conference, are called *Concurrency: Theory, Language, And Architecture* (1989: Oxford, UK), *Concurrency* (1988: Hamburg, Germany), and *Seminar on Concurrency* (1984: Pittsburgh, PA, USA), respectively.

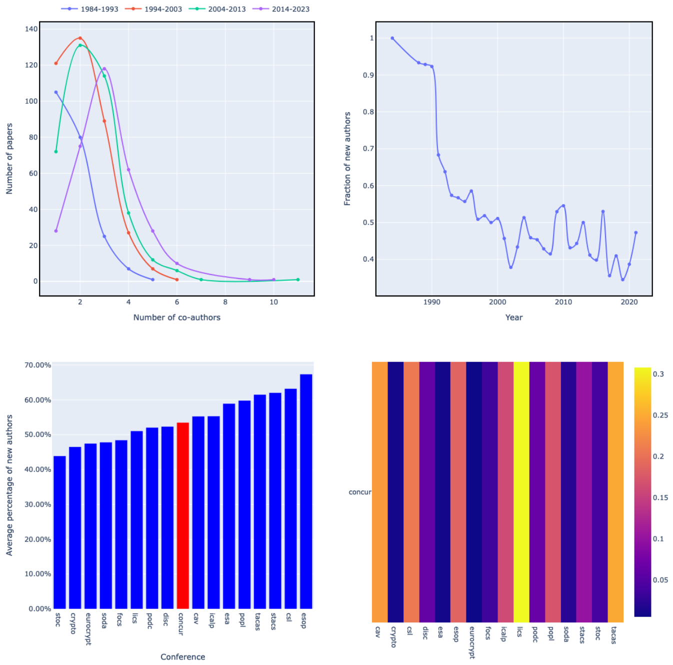


Figure 2: The evolution of the number of co-authors per decade (top left), the percentage of new authors per year (top right), the average number of new authors compared with 16 theoretical computer science conferences (bottom left), and the values of the Sørensen-Dice similarity index with respect to the same 16 conferences (bottom right).

38), the number of authors more than doubled (from 52 to 110). This is probably justified by the fact that the number of co-authors per paper has increased significantly over the years, as it is shown in the top left part of Figure 2. Indeed, while in the first decade the number of papers with a single author was the majority and the maximum number of co-authors was five, in the last decade the papers with two, three, and even four authors have become more popular than single-author papers. At the same time, the maximum number of co-authors has increased to ten. As indicated by a similar data- and graph-mining analysis for ICALP and other major conferences in theoretical computer science reported at `https://slides.com/piluc/icalp-50?token=fl3BBJ8j#/2/5`, papers authored by two to four researchers are now more frequent than singly-authored ones in all fields of the theory of computing.

The top right part of Figure 2 shows the evolution of the percentage of *new* distinct authors of the published papers per year. This percentage decreased and stabilized between 40% and

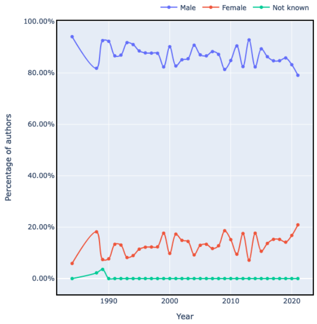


Figure 3: The evolution of the percentage of male and female authors per year (the two percentages are computed with respect to the number of authors for which the sex has been assigned). The percentage of authors with no sex assigned is also shown (with respect to the total number of authors).

50%. In other words, every year approximately half of the authors of the CONCUR conference are new authors. (Note that, in this analysis, we are not considering the co-authorship between authors, that is, we are not verifying whether the new authors have been "introduced" by an author who already published in the conference.) The percentage of new authors for several conferences in theoretical computer science is available at `https://slides.com/piluc/icalp-50?token=fl3BBJ8j#/2/3`. We find it noteworthy that the percentage of new authors for 11 of the conferences considered in that plot is above 50% (see also the bottom left part of Figure 2, where the bar corresponding to CONCUR is shown in red).

Finally, the bottom right part of the figure shows the values of the Sørensen-Dice index of similarity computed by comparing the set of CONCUR authors with the sets of authors for sixteen theoretical computer science conferences[6]. As it can be seen, the conference that is most similar to CONCUR is LICS (with Sørensen-Dice index approximately equal to 0.3), followed by TACAS (approximately 0.25), CAV (approximately 0.24), and CSL (approximately 0.21). The least similar conferences to CONCUR are, instead, EUROCRYPT, ESA, and CRYPTO (all below 0.01).

4 Sex analysis

The sex of CONCUR authors has been determined mostly by querying the web service available at `genderize.io` (which is based on first names only), and partly by manually searching

[6] Given two sets A and B, the Jaccard index $J(A, B)$ is equal to $\frac{|A\cap B|}{|A\cup B|}$, and the Sørensen-Dice index is equal to $\frac{2J(A,B)}{1+J(A,B)}$ (see T. Sørensen, "A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on Danish commons", *Kongelige Danske Videnskabernes Selskab.*, 5 (4): 1–34 (1948), and L.R. Dice, "Measures of the Amount of Ecologic Association Between Species", *Ecology*, 26 (3): 297–302 (1945)).

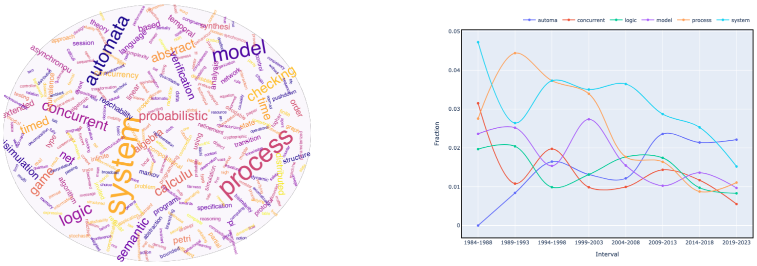


Figure 4: The word cloud corresponding to the words contained in the titles of CONCUR papers (left) and the evolution of fractions of occurrences per five-year interval of the six words globally most frequent (right).

the authors on the web. At the end of this phase, almost all authors have been assigned a sex (which should not be confused with their gender—see, for instance, the interview with Judith Butler at `https://www.youtube.com/watch?v=Bo7o2LYATDc`). Figure 3 shows the evolution of the percentages of male and female authors per year (the percentage of authors with no sex assigned is also shown). The percentage of female authors increased from approximately 6% to approximately 21% over the years. However, the number of women is still approximately only one fifth of the total number of authors, which maybe indicates that some reflections have to be done on this subject[7]. Note, however, that, as indicated by the data displayed at `https://slides.com/piluc/icalp-50?token=fl3BBJ8j#/3/1`, these numbers are consistent with the ones of many other theoretical computer science conferences, where the percentage of female authors was below 20% in 2021.

5 Topic analysis

The word cloud corresponding to the words contained in the titles of CONCUR papers is shown in the left part of Figure 4. As it can be seen, the words `automa`, `concurrent`, `logic`, `model`, `process`, and `system` are those that appear more frequently in the title of a CONCUR paper.

Of all the words contained in the titles of CONCUR papers in a certain time interval, the plot on the right part of Figure 4 shows what fraction of them are one of the above most frequent six words. It can be seen that `system` is almost always the most frequent one, while the other five words alternate and three of them have been the most frequent one in at least one time

[7] As mentioned in the recently published opinion article available at `https://www.scientificamerican.com/article/there-are-too-few-women-in-computer-science-and-engineering/`, which summarizes the main findings in the paper Allison Master, Andrew N. Meltzoff, and Sapna Cheryan, "Gender stereotypes about interests start early and cause gender disparities in computer science and engineering", *Proceedings of the National Academy of Sciences* 118 (48) e2100030118 (2021), `https://www.pnas.org/content/118/48/e2100030118`, sex-based stereotypes related to computer science and engineering seem to become entrenched early in life. Indeed, as reported in those studies, children and adolescents in the U.S. already believe that girls are less interested than boys in computer science and engineering. Experiments reported in the above-mentioned PNAS paper indicate that the culture in computer science and engineering contributes to excluding girls and women.

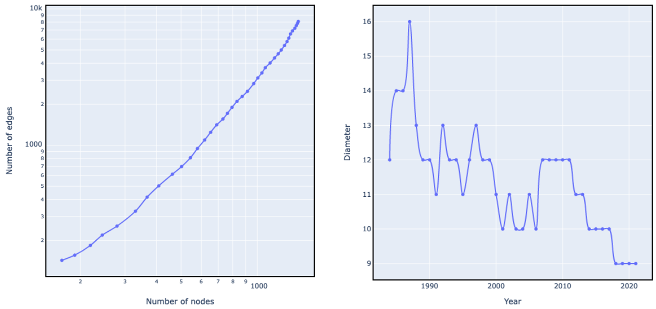


Figure 5: The densification (left) and the diameter shrinking (right) of the collaboration graph of CONCUR authors.

interval. The interested reader can see the evolution of all the words appearing in the title of some CONCUR paper at the web page `http://www.pilucrescenzi.it/concur/word_frequencies_5.html`, where it is also possible to compare the evolution of two different words.

6 Basic graph mining

The *static graph* (or collaboration graph) of CONCUR is an undirected graph whose nodes are the authors who presented at least one paper at CONCUR, and whose edges (a_1, a_2) correspond to two authors a_1 and a_2 who co-authored at least one paper (not necessarily presented at CONCUR). In other words, this graph is the subgraph of the DBLP graph induced by the set of CONCUR authors.

The static graph has 1451 nodes and 8086 edges. It is a sparse graph, since its density[8] is approximately equal to 0.008. It contains a giant connected component, which includes approximately 98% of all nodes.

Two phenomena that have been pointed out in the literature are the *densification* of a social network and the *shrinking* of its diameter[9]. In Figure 5, these two phenomena are represented in the left and the right part of the figure, respectively. Indeed, it can be seen how the number of edges increases more than linearly with respect to the number of nodes, and that the diameter decreases from 12 to 9 (even if the number of nodes increases).

We also compute the evolution of the degrees of separation, that is, the average distance

[8] The density of an undirected graph with n nodes and m edges is $\frac{2m}{n(n-1)}$, that is, the ratio of its number of edges with respect to the maximum number of possible edges. For a definition of most of the notions used in this section and in the next one and for a description of the used algorithms, we refer the interested reader to the lecture notes available at `https://github.com/piluc/GraphMining`.

[9] See J. Leskovec, J.M. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters", *ACM Trans. Knowl. Discov. Data*, 1:1, 2 (2007).

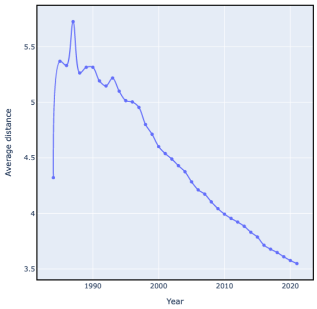


Figure 6: The evolution of the degrees of separation of the collaboration graph of CONCUR authors.

between any two authors in the largest connected component[10]. This evolution (which is similar to the evolution of the diameter) is shown in Figure 6. As it can be seen, the CONCUR community is quite a small world, in which the average distance is currently approximately 3.5.

7 Centrality measures

Centrality measures are a key tool for understanding social networks and are used to assess the "importance" of a given node[11]. In order to quantify the role played by CONCUR authors, we compute the following three different centrality measures on the largest connected component of the static graph.

Degree This is the number of neighbors (that is the number of coauthors).

Closeness This is the average distance from one author to all other authors of its connected component.

Betweenness This is the fraction of shortest paths, passing through one author, between any pair of other authors in its connected component.

In Table 1, we show the top ten CONCUR authors with respect to the above-mentioned three centrality measures in decreasing order. As expected, several authors appear in multiple lists: this is due to the well-known phenomenon of correlation between the centrality measures. It is also interesting to observe that the two female scientists included in the lists, namely Marta Z. Kwiatkowska and Catuscia Palamidessi, appear in the closeness and the betweenness lists. This indicates that they maybe do have fewer coauthors than other "central colleagues", but

[10] The study of the degrees of separation and of the so-called *small-world phenomenon* started with the experiment described in S. Milgram, "The Small World Problem", *Psychology Today*, 1:1, 61–67 (1967).

[11] See L.C. Freeman, "Centrality in social networks conceptual clarification", *Social Networks*, 1, 215—239 (1978).

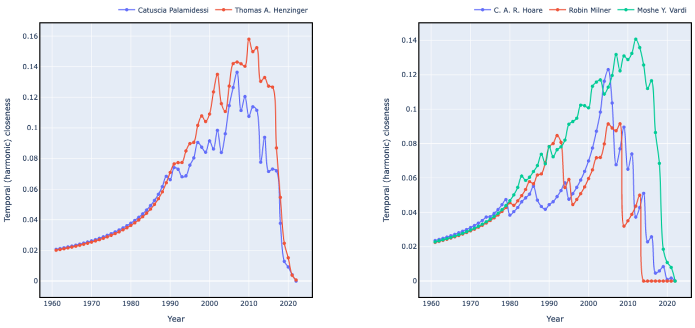


Figure 7: The evolution of the temporal harmonic closeness of Thomas A. Henzinger and Catuscia Palamidessi (left) and of Tony Hoare, Robin Milner, and Moshe Y. Vardi (right).

that their collaborations make them either quite close to the rest of the community or a sort of "bridge". Finally, it might be interesting to determine the centrality of an author by analysing the citation network. However, this network cannot be easily and precisely derived by using only the DBLP data, and other data repositories should be used (such as, for instance, the OpenAlex service available at `https://openalex.org/`).

Degree	Closeness	Betweenness
Thomas A. Henzinger	Kim G. Larsen	Kim G. Larsen
Kim G. Larsen	Moshe Y. Vardi	Thomas A. Henzinger
Moshe Y. Vardi	Thomas A. Henzinger	Moshe Y. Vardi
Axel Legay	Axel Legay	Javier Esparza
James Worrell	Joost-Pieter Katoen	Catuscia Palamidessi
Krishnendu Chatterjee	Luca Aceto	Axel Legay
Joost-Pieter Katoen	Javier Esparza	Joost-Pieter Katoen
Rupak Majumdar	Marta Z. Kwiatkowska	Luca Aceto
Jean-François Raskin	Catuscia Palamidessi	Rupak Majumdar
Javier Esparza	Rupak Majumdar	Scott A. Smolka

Table 1: The top-10 CONCUR authors with respect to three centrality measures

7.1 Temporal closeness

The *temporal graph* has the same set of nodes of the static graph, but the edges (a_1, a_2, y) correspond to two authors a_1 and a_2 who co-authored in year y at least one paper (not necessarily presented at CONCUR). In the case of this graph, we compute the *temporal closeness*, which is intuitively the area covered by the plot of the temporal harmonic closeness of an author[12].

[12] The *temporal harmonic closeness* of a node u at time t is defined as $\frac{1}{n-1}\sum_{v\neq u}\frac{1}{d_t(u,v)}$, where $d_t(u,v)$ is the time duration of the earliest arrival path starting no earlier than t (see P. Crescenzi, C. Magnien, and A. Marino,

For example, in the left part of Figure 7, the plot of the temporal harmonic closeness of Thomas A. Henzinger and of Catuscia Palamidessi are shown, while the right part depicts the temporal harmonic closeness of Tony Hoare, Robin Milner, and Moshe Y. Vardi. By computing the area covered by these two plots, we may conclude that the temporal closeness of Henzinger is higher than Palamidessi's one. The top ten CONCUR authors with respect to this centrality measure are Moshe Y. Vardi, Kim G. Larsen, Thomas A. Henzinger, Joost-Pieter Katoen, Javier Esparza, Orna Kupferman, Edmund M. Clarke, Ugo Montanari, Rocco De Nicola, and Marta Z. Kwiatkowska.

Several other notions of temporal centrality have been introduced in the literature in the last few years. For instance, the temporal analogue of the betweenness centrality has been deeply analyzed and, since such a measure cannot be efficiently computed even in the case of medium-sized graphs, approximation algorithms based on sampling techniques have been proposed[13]. We believe that it would be interesting to apply these algorithms to the temporal graph of the CONCUR collaborations.

"Finding Top-*k* Nodes for Temporal Closeness in Large Temporal Graphs", *Algorithms*, 13:9, 211, (2020)). Note that in a temporal graph a path is a sequence of edges such that each edge appears later than the edges preceding it.

[13] See S. Buß, H. Molter, R. Niedermeier, and M. Rymar, "Algorithmic Aspects of Temporal Betweenness", *KDD*, 2084–2092 (2020), and D. Santoro and I. Sarpe, "ONBRA: Rigorous Estimation of the Temporal Betweenness Centrality in Temporal Networks", *WWW*, 1579–1588, (2022).

Report on NCMA 2022:
12th International Workshop on Non-Classical Models of Automata and Applications
Debrecen, Hungary, August 26–27, 2022

Bianca Truthe
Justus Liebig University Giessen, Germany

The workshop series on Non-Classical Models of Automata and Applications (NCMA) was founded in the year 2009; up to the year 2019, it took place every year. After a pandemic caused pause, there was a restart with a new edition this year. The history and other information can be found at the homepage of the workshop series which is available here:

`https://www.cs.uni-potsdam.de/NCMA/`

The 12th edition was held in Debrecen (Hungary) on August 26 and 27, 2022. The workshop was organized by György Vaszil and his colleagues from the University of Debrecen as well as Henning Bordihn from the University of Potsdam. It was co-located with the conferences DCFS and MCU.

At the conference venue

The invited speakers were

- Florin Manea (University of Göttingen, Germany) who gave a survey on ‘Combinatorial Algorithms for Subsequence Matching’ and
- Gyula Klima (Fordham University, NY, USA) with a talk on ‘Language and Intelligence, Artificial vs. Natural, or What can and what cannot AI do with Natural Language’.

Besides the talks by the invited speakers, 15 talks on peer reviewed research papers were presented. The 10 regular papers and 5 short papers covered many topics in the area of automata theory and beyond (Non-returning finite automata, involutory automata, P systems, permutation automata, counter automata, reaction systems, typewriter automata, Watson-Crick automata, and many more).

On the workshop website, you can find the program with all talks:

`https://konferencia.unideb.hu/en/ncma-2022`

The proceedings were edited by Henning Bordihn, Géza Horváth, and György Vaszil and published in the EPTCS series (Electronic Proceedings in Theoretical Computer Science, `https://eptcs.org/`), Volume 367. Extended versions of selected papers will be published in a special issue of *RAIRO – Theoretical Informatics and Applications*.

The social program on the first day, after the business meeting, consisted of a guided tour in the city center of Debrecen and a Dinner afterwards in a nice restaurant with a variety of delicious meals.

Many thanks to the organizers, program committee members, external reviewers, and participants for the pleasant and successful event. The next issue of NCMA is planned to be held next year in Famagusta (North Cyprus), co-located with CIAA, organized by Benedek Nagy. We invite all readers of this report to submit papers to NCMA 2023 and to come to the Mediterranean Sea in next September.

Report on DCFS 2022:
24th International Conference on Descriptional Complexity of Formal Systems
Debrecen, Hungary, August 29–31, 2022

Bianca Truthe
Justus Liebig University Giessen, Germany

The 24th DCFS took place in Debrecen, Hungary, from August 29 to 31, after the NCMA and before the MCU at the same place. It was organized jointly by the IFIP Working Group 1.02 on Descriptional Complexity and by the Faculty of Informatics at the University of Debrecen.

Debrecen City Center

At the conference, 18 scientific talks were given, 4 of them by invited speakers, namely

- Galina Jirasková (Slovak Academy of Sciences, Košice, Slovakia) who spoke about 'Operations on unambiguous finite automata' on the first day,
- Mikołaj Bojańczyk (University of Warszaw, Poland) with a talk on 'Polyregular functions' in the morning of the second day,
- Szabolcs Iván (University of Szeged, Hungary) who spoke about 'Scattered context-free order types' in the afternoon of the second day, and
- Stefano Crespi Reghizzi (Polytechnic University of Milan, Italy) who gave a talk on 'The alphabetic complexity in homomorphic definitions of word, tree, and picture languages' on the third day.

The other 14 contributions (all peer reviewed) were written by 33 authors. All papers are contained in the proceedings, edited by Yo-Sub Han and György Vaszil, and published by Springer as volume 13439 in the series *Lecture Notes in Computer Science*. Full versions of selected papers will be published in a special issue of the journal *Theoretical Computer Science*.

On the workshop website, you can find the program with all talks:

`https://konferencia.unideb.hu/en/dcfs-2022`

Besides the scientific sessions, there were two more: the Business Meeting of the IFIP Working Group 1.02 and a Special Session to honour four scientists who passed away in the last three years and who had a strong connection to the research area of this conference. In four emotional presentations, colleagues and friends honoured Janusz Brzozowski (presented by Rogério Reis), Helmut Jürgensen (by Henning Bordihn), Alica Kelemenová (by Erzsébet Csuhaj-Varjú), and Detlef Wotschke (by Andreas Malcher).

After the Business Meeting where Martin Kutrib as the chair of the working group gave an overview about activities of the group as well as the past and future of the conference series DCFS, a social meeting followed which consisted of a visit at the Center for Modern and Contemporary Art and a great conference dinner.

The history of the conference series DCFS and other information can be found at the homepage:

`http://www.informatik.uni-giessen.de/dcfs/`

We thank everybody, in particular the local organizers, who made the conference a successful event. The next DCFS will take place in Potsdam, Germany, organized by Henning Bordihn. We invite all readers of this report to submit papers to DCFS 2023 and to come to Potsdam in next July.

Report on MCU 2022:
9th International Conference on Machines, Computations, and Universality
Debrecen, Hungary, August 31 – September 2, 2022

Bianca Truthe
Justus Liebig University Giessen, Germany

After NCMA and DCFS, the 9th MCU was the third conference in a row which took place in Debrecen, Hungary (August 31 to September 2). Also this conference was organized by György Vaszil and his colleagues from the Faculty of Informatics at the University of Debrecen.

Debrecen Big Forest Park

Besides the talks of the four invited speakers, the scientific program consisted of 10 talks presenting peer reviewed research papers. The four invited speakers were

- Hava Siegelmann (University of Massachusetts Amherst, USA) with a talk on ‘Super Turing computing enables lifelong learning AI’ on the first day,
- Mika Hirvensalo (University of Turku, Finland) who spoke about ‘Using inference to boost computing’ in the morning of the second day,
- Bianca Truthe (University of Gießen, Germany) who gave ‘A survey on computationally complete accepting and generating networks of evolutionary processors’ in the afternoon of the second day, and
- Enrico Formenti (University of Côte d’Azur, Sophia Antipolis, France) with a talk on ‘Complexity of local, global and universality properties in finite dynamical systems’ on the third day.

The accepted regular papers were written by 24 authors and have all been peer reviewed. The proceedings with all these papers were edited by Jérôme Durand-Lose and György Vaszil and published by Springer as volume 13419 in the series

Lecture Notes in Computer Science. Full versions of selected papers will be published in a special issue of the *International Journal of Foundations of Computer Science*.

On the workshop website, you can find the program with all talks:

`https://konferencia.unideb.hu/en/mcu-2022`

The Program Committee agreed on the following awards: The Best Paper Award went to Lucie Ciencialová, Ludek Cienciala, and Erzsébet Csuhaj-Varjú for their paper 'Languages of Distributed Reaction Systems'. The Best Student Paper Award was given to Manon Blanc and Olivier Bournez for their paper 'A characterization of polynomial time computable functions from the integers to the reals using discrete ordinary differential equations'.

The social meeting consisted of a visit at the Center for Modern and Contemporary Art (another exhibition than the one we had visited during DCFS) and a great conference dinner again.

On the last day of the conference, Jérôme Durand-Lose announced the new composition of the Steering Committee, after some members had expressed their wish to step down and future members were invited to join. The new Steering Committee consists of Erzsébet Csuhaj-Varjú, Jérôme Durand-Lose (Chair), Rudolf Freund, Daniela Genova, Maurice Margenstern, Benedek Nagy, Alberto Ottavio Leporati, Shinnosuke Seki, Bianca Truthe, György Vaszil, and Sergey Verlan.

Participants at the conference venue

We thank all participants, program committee members, external reviewers, and especially the organizers for the successful conference. The next issue of MCU is planned to be held in two years in Nice (France).

Report on CPM 2022

The 33rd Annual Symposium on Combinatorial Patter Matching

Nadia Pisanti
University of Pisa, Italy

The 33rd Annual Symposium on Combinatorial Patter Matching was held in Prague, Czech Republic, from June 27th to June 29th, 2022. Each year, CPM gathers scientists of many areas related to word combinatorics, discrete algorithms, string algorithms, that address problems such as text searching and indexing, data compression, pattern discovery, and that can be applied to bioinformatics, data mining, information retrieval, natural language processing, just to mention a few. The 2022 edition of CPM was held at the Faculty of Civil Engineering of the Czech Technical University in Prague, organised by Jan Holub, Jan Trávníček, Ondřej Guth, Tomáš Pecka, and Eliška Šestáková, Dominika Draesslerová, Štepán Plach y, Lucie Procházková, Regina Šmidová. The scientific program consisted of 3 invited talks, 2 highlights talks, and 26 regular talks of accepted papers which had been chosen by the Program Committee out of 43 submissions (coming from authors from 20 different countries and 4 different continents) on the basis of three reviews for each submission. The Program Committee consisted of 28 members (24 men, 4 women) from 18 different countries.
The detailed program, as well as some pictures can be found on the website https://www.stringology.org/event/CPM2022/.
The proceedings, edited by the program committee co-chairs Hideo Bannai and Jan Holub, have been published in volume 223 of LIPIcs. They are open access and can be found here:
https://drops.dagstuhl.de/opus/portals/lipics/index.php?semnr=16232.
The invited talks covered several interesting topics and were given by:

1. ***Takehiro Ito*** (Tohoku University, Japan):
   "Invitation to Combinatorial Reconfiguration"
2. ***Jeffrey Shallit*** (University of Waterloo, Canada):
   "Using automata and a decision procedure to prove results in pattern matching"
3. ***Sharma V. Thankachan*** (University of Central Florida, USA):
   "Compact Text Indexing for Advanced Pattern Matching Problems: Parametrized, Order-isomorphic, 2D, etc."

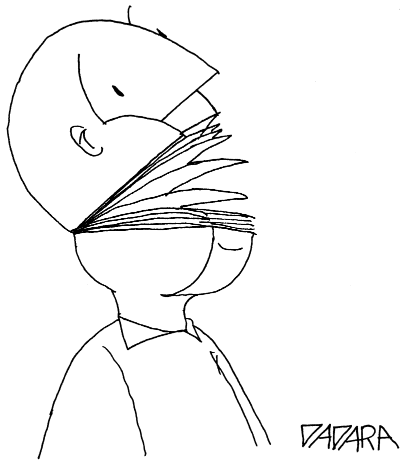
The highlights talks, introduced for the first time in CPM 2019, are special sessions dedicated to as many presentations of the highlights of recent results and

developments in combinatorial pattern matching topics, that have been recently published in other venues.
This year, CPM featured the following two highlight talks:

1. ***Tomasz Kociumaka*** ((MPI, Germany). "Small space and streaming pattern matching with k edits", paper presented at FOCS 2021.
2. ***Moses Ganardi*** (MPI, Germany). "Compression by Contracting Straight-Line Programs", paper presented at ESA 2021, and extended in J.ACM 2021.

The conference had 50 on-line and 59 in-person participants. The business meeting of CPM 2022 was chaired by the Steering Commitee and took place on June 27th, at the end of the afternoon session. In the business meeting, the PC chair and local organiser Jan Holub gave briefly an overview on the conference organisation. At the end of the meeting, Laurent Bulteau presented the conference edition of CPM 2023 which will take place in Paris, France. The social program took place the second day of CPM 2022 and started with a sightseeing ride through the city center on board of one of Prague's historical trams that used to roam Prague streets in the first half of the 20th century. The tram ride was followed by a guided tour through beautiful Prague's Old Town which started with the Charles Bridge and ended with a conference dinner at the Tiskarna Restaurant.
Thank you to the local organizers for their excellent work and all participants for the nice and successful conference.
Looking forward to see you at CPM 2023 in Paris!

Contributions by EATCS Award Recipients



Interviews with the 2022 CONCUR Test-of-Time Award Recipients

Luca Aceto
ICE-TCS, Department of Computer Science,
Reykjavik University
Gran Sasso Science Institute, L'Aquila
luca@ru.is, luca.aceto@gssi.it

Orna Kupferman
School of Computer Science and Engineering
Hebrew University, Jerusalem
orna@cs.huji.ac.il

Mickael Randour
Faculty of Science, Mathematics Department
Université de Mons
mickael.randour@gmail.com

Davide Sangiorgi
Department of Computer Science, University of Bologna
Davide.Sangiorgi@cs.unibo.it

In 2020, the CONCUR conference series instituted its Test-of-Time Award, whose purpose is to recognise important achievements in Concurrency Theory that were published at the CONCUR conference and have stood the test of time. This year, the following four papers were chosen to receive the CONCUR Test-of-Time Awards for the periods 1998–2001 and 2000–2003 by a jury consisting of Ilaria Castellani (chair), Paul Gastin, Orna Kupferman, Mickael Randour and Davide Sangiorgi. (The papers are listed in chronological order.)

- Christel Baier, Joost-Pieter Katoen and Holger Hermanns. Approximate symbolic model checking of continuous-time Markov chains. CONCUR 1999.

- Franck Cassez and Kim Guldstrand Larsen. The Impressive Power of Stopwatches. CONCUR 2000.
- James J. Leifer and Robin Milner. Deriving Bisimulation Congruences for Reactive Systems. CONCUR 2000.
- Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga. The Element of Surprise in Timed Games. CONCUR 2003.

This article is devoted to interviews with the recipients of the Test-of-Time Award. More precisely,

- Orna Kupferman interviewed Christel Baier, Joost-Pieter Katoen and Holger Hermanns;
- Luca Aceto interviewed Franck Cassez and Kim Guldstrand Larsen;
- Davide Sangiorgi interviewed James Leifer; and
- Luca Aceto and Mickael Randour jointly interviewed Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga.

We are very grateful to the awardees for their willingness to answer our questions and hope that the readers of this article will enjoy reading the interviews as much as we did.

Interview with C. Baier, J.-P. Katoen and H. Hermanns

In what follows, BHK refers to Baier, Katoen and Hermanns.

Orna: You receive the CONCUR Test-of-Time Award 2022 for your paper "Approximate symbolic model checking of continuous-time Markov chains," which appeared at CONCUR 1998[1]. In that article, you combine three different challenges: symbolic algorithms, real-time systems, and probabilistic systems. Could you briefly explain to our readers what the main challenge in such a combination is?

BHK: The main challenge is to provide a fixed-point characterization of time-bounded reachability probabilities: the probability to reach a given target state

[1] See `https://link.springer.com/content/pdf/10.1007/3-540-48320-9_12.pdf`.

within a given deadline. Almost all works in the field up to 1999 treated discrete-time probabilistic models and focused on "just" reachability probabilities: what is the probability to eventually end up in a given target state? This can be characterized as a unique solution of a linear equation system. The question at stake was: how to incorporate a real-valued deadline d? The main insight was to split the problem in staying a certain amount of time, x say, in the current state and using the remaining $d - x$ time to reach the target from its successor state. This yields a Volterra integral equation system; indeed time-bounded reachability probabilities are unique solutions of such equation systems. In the CONCUR 1999 paper we suggested to use symbolic data structures to do the numerical integration; later we found out that much more efficient techniques can be applied.

Orna: Could you tell us how you started your collaboration on the award-winning paper? In particular, as the paper combines three different challenges, is it the case that each of you has brought to the research different expertise?

BHK: Christel and Joost-Pieter were both in Birmingham, where a meeting of a collaboration project between German and British research groups on stochastic systems and process algebra took place. There the first ideas of model checking continuous-time Markov chains arose, especially for time-bounded reachability: with stochastic process algebras there were means to model CTMCs in a compositional manner, but verification was lacking. Back in Germany, Holger suggested to include a steady-state operator, the counterpart of transient properties that can be expressed using timed reachability probabilities. We then also developed the symbolic data structure to support the verification of the entire logic.

Orna: Your contribution included a generalization of BDDs (binary decision diagrams) to MTDDs (multi-terminal decision diagrams), which allow both Boolean and real-valued variables. What do you think about the current state of symbolic algorithms, in particular the choice between SAT-based methods and methods that are based on decision diagrams?

BHK: BDD-based techniques entered probabilistic model checking in the mid 1990's for discrete-time models such as Markov chains. Our paper was one of the first, perhaps even the first, that proposed to use BDD structures for real-time stochastic processes. Nowadays, SAT and in particular SMT-based techniques belong to the standard machinery in probabilistic model checking. SMT techniques are, e.g., used in bisimulation minimization at the language level, counterexample generation, and parameter synthesis. This includes both linear as well as non-linear theories. BDD techniques are still used, mostly in combination with sparse representations, but it is fair to say that SMT is becoming more and more relevant.

Orna: What are the research topics that you find most interesting right now? Is there any specific problem in your current field of interest that you'd like to see solved?

BHK: This depends a bit on whom you ask! Christel's recent work is about cause-effect reasoning and notions of responsibility in the verification context. This ties into the research interest of Holger who looks at the foundations of perspicuous software systems. This research is rooted in the observation that the explosion of opportunities for software-driven innovations comes with an implosion of human opportunities and capabilities to understand and control these innovations. Joost-Pieter focuses on pushing the borders of automation in weakest-precondition reasoning of probabilistic programs. This involves loop invariant synthesis, probabilistic termination proofs, the development of deductive verifiers, and so forth. Challenges are to come up with good techniques for synthesizing quantitative loop invariants, or even complete probabilistic programs.

Orna: What advice would you give to a young researcher who is keen to start working on topics related to symbolic algorithms, real-time systems, and probabilistic systems?

BHK: Try to keep it smart and simple.

Interview with Franck Cassez and Kim Guldstrand Larsen

Luca: You receive the CONCUR Test-of-Time Award 2022 for your paper "The Impressive Power of Stopwatches"[2], which appeared at CONCUR 2000. In that article, you showed that timed automata enriched with stopwatches and unobservable time delays have the same expressive power of linear hybrid automata. Could you briefly explain to our readers what timed automata with stopwatches are? Could you also tell us how you came to study the question addressed in your award-winning article? Which of the results in your paper did you find most surprising or challenging?

Kim: Well, in timed automata all clocks grow with rate 1 in all locations of the automata. Thus you can tell the amount of time that has elapsed since a particular clock was last reset, e.g., due to an external event of interest. A stopwatch is a real-valued variable similar to a regular clock. In contrast to a clock, a stopwatch will in certain locations grow with rate 1 and in other locations grow with rate 0, i.e., it is stopped. As such, a stopwatch gives you information about the accumulated time spent in a certain parts of the automata.

In modelling schedulability problems for real-time systems, the use of stopwatches is crucial in order to adequately capture preemption. I definitely believe

[2] See `https://link.springer.com/content/pdf/10.1007/3-540-44618-4_12.pdf`.

that it was our shared interest in schedulability that brought us to study timed automata with stopwatches. We knew from earlier results by Alur et al. that properties such as reachability was undecidable. But what could we do about this? And how much expressive power would the addition of stopwatches provide?

In the paper we certainly put the most emphasis on the latter question, in that we showed that stopwatch automata and linear hybrid automata accept the same class of timed languages, and this was at least for me the most surprising and challenging result. However, focusing on impact, I think the approximate zone-based method that we apply in the paper has been extremely important from the point of view of having our verification tool UPPAAL being taken up at large by the embedded systems community. It has been really interesting to see how well the over-approximation method actually works.

Luca: In your article, you showed that linear hybrid automata and stopwatch automata accept the same class of timed languages. Would this result still hold if all delays were observable? Do the two models have the same expressive power with respect to finer notions of equivalence such as timed bisimilarity, say? Did you, or any other colleague, study that problem, assuming that it is an interesting one?

Kim: These are definitely very interesting questions, and should be studied. As for finer notions of equivalences, e.g., timed bisimilarity, I believe that our translation could be shown to be correct up to some timed variant of chunk-by-chunk simulation introduced by Anders Gammelgaard in his Licentiat Thesis from Aarhus University in 1991[3]. That could be a good starting point.

Luca: Did any of your subsequent research build explicitly on the results and the techniques you developed in your award-winning paper? Which of your subsequent results on timed and hybrid automata do you like best? Is there any result obtained by other researchers that builds on your work and that you like in particular or found surprising?

Kim: Looking up in DBLP, I see that I have some 28 papers containing the word "scheduling". For sure stopwatches will have been used in one way or another in these. One thing that we never really examined thoroughly is to investigate how well the approximate zone-based technique will work when applied to the translation of linear hybrid automata into stopwatch automata. This would definitely be interesting to find out.

This was the first joint publication between me and Franck. I enjoyed fully the collaboration on all the next 10 joint papers. Here the most significant ones are probably the paper at CONCUR 2005, where we presented the symbolic on-the-fly algorithms for synthesis for timed games and the branch UPPAAL TIGA. And

[3] See `https://tidsskrift.dk/daimipb/article/view/6611/5733`.

later in a European project GASICS with Jean-Francois Raskin, we used TIGA in the synthesis of optimal and robust control of a hydraulic system.

Franck: Using the result in our paper, we can analyse scheduling problems where tasks can be stopped and restarted, using real-time model-checking and a tool like UPPAAL.

To do so, we build a network of stopwatch automata modelling the set of tasks and a scheduling policy, and reduce schedulability to a safety verification problem: avoid reaching states where tasks do not meet their deadlines. Because we over-approximate the state space, our analysis may yield some false positives and may wrongly declare a set of tasks non-schedulable because the over-approximation is too coarse.

In the period 2003–2005, in cooperation with Francois Laroussinie we tried to identify some classes of stopwatch automata for which the over-approximation does not generate false positives. We never managed to find an interesting subclass.

This may look like a serious problem in terms of applicability of our result, but in practice, it does not matter too much. Most of the time, we are interested in the schedulability of a specific set of tasks (e.g., controlling a plant, a car, etc.) and for these instances, we can use our result: if we have false positives, we can refine the model tasks and scheduler and rule them out. Hopefully after a few iterations of refinement, we can prove that the set of tasks is schedulable.

The subsequent result on timed and hybrid automata of mine that I probably like best is the one we obtained on solving optimal reachability in timed automata. We had a paper at FSTTCS in 2004[4] presenting the theoretical results, and a companion paper at GDV 2004[5] with an implementation using HyTech, a tool for analysing hybrid automata.

I like these results because we ended up with a rather simple proof, after 3–4 years working on this hard problem.

Luca: Could you tell us how you started your collaboration on the award-winning paper? I recall that Franck was a regular visitor to our department at Aalborg University for some time, but I can't recall how his collaboration with the UPPAAL group started.

Kim: I am not quite sure I remember how and when I first met Franck. For some time we already worked substantially with French researchers, in particular from LSV Cachan (Francois Larroussinie and Patricia Bouyer). I have the feeling that there were quite some strong links between Nantes (were Franck was) and LSV on timed systems in those days. Also Nantes was the organizer of the PhD school

[4] See https://doi.org/10.1007/978-3-540-30538-5_13.

[5] See https://doi.org/10.1016/j.entcs.2004.07.006.

MOVEP five times in the period 1994-2002, and I was lecturing there in one of the years, meeting Olivier Roux and Franck who were the organizers. Funny enough, this year we are organizing MOVEP in Aalborg. Anyway, at some point Franck became a regular visitor to Aalborg, often for long periods of time—playing on the squash team of the city when he was not working.

Franck: As Kim mentioned, I was in Nantes at that time, but I was working with Francois Laroussinie who was in Cachan. Francois had spent some time in Aalborg working with Kim and his group and he helped organise a mini workshop with Kim in 1999, in Nantes. That's when Kim invited me to spend some time in Aalborg, and I visited Aalborg University for the first time from October 1999 until December 1999. This is when we worked on the stopwatch automata paper. We wanted to use UPPAAL to verify systems beyond timed automata.

I visited Kim and his group almost every year from 1999 until 2007, when I moved to Australia. There were always lots of visitors at Aalborg University and I was very fortunate to be there and learn from the Masters.

I always felt at home at Aalborg University, and loved all my visits there. The only downside was that I never managed to defeat Kim at badminton. I thought it was a gear issue, but Kim gave me his racket (I still have it) and the score did not change much.

Luca: What are the research topics that you find most interesting right now? Is there any specific problem in your current field of interest that you'd like to see solved?

Kim: Currently I am spending quite some time on marrying symbolic synthesis with reinforcement learning for Timed Markov Decision Processes in order to achieve optimal as well as safe strategies for Cyber-Physical Systems.

Luca: Both Franck and you have a very strong track record in developing theoretical results and in applying them to real-life problems. In my, admittedly biased, opinion, your work exemplifies Ben Schneiderman's Twin-Win Model[6], which propounds the pursuit of "the dual goals of breakthrough theories in published papers and validated solutions that are ready for widespread dissemination." Could you say a few words on your research philosophy?

Kim: I completely subscribe to this. Several early theoretical findings, such as the paper on stopwatch automata, have been key in our sustainable transfer to industry.

Franck: Kim has been a mentor to me for a number of years now, and I certainly learned this approach/philosophy from him and his group.

[6] See https://www.pnas.org/doi/pdf/10.1073/pnas.1802918115.

We always started from a concrete problem, e.g., scheduling tasks/checking schedulability, and to validate the solutions, building a tool to demonstrate applicability. The next step was to improve the tool to solve larger and larger problems.

UPPAAL is a fantastic example of this philosophy: the reachability problem for timed automata is PSPACE-complete. That would deter a number of people to try and build tools to solve this problem. But with smart abstractions, algorithms and data-structures, and constant improvement over a number of years, UPPAAL can analyse very large and complex systems. It is amazing to see how UPPAAL is used in several areas from traffic control to planning and to precisely guiding a needle for an injection.

Luca: What advice would you give to a young researcher who is keen to start working on topics related to formal methods?

Kim: Come to Aalborg, and participate in next year's MOVEP.

Interview with James Leifer

Davide: How did the work presented in your CONCUR Test-of-Time paper come about?

James: I was introduced to Robin Milner by my undergraduate advisor Bernard Sufrin around 1994. Thanks to that meeting, I started with Robin at Cambridge in 1995 as a fresh Ph.D. student. Robin had recently moved from Edinburgh and had a wonderful research group, including, at various times, Peter Sewell, Adriana Compagnoni, Benjamin Pierce, and Philippa Gardner. There were also many colleagues working or visiting Cambridge interested in process calculi: Davide Sangiorgi, Andy Gordon, Luca Cardelli, Martín Abadi,. . . . It was an exciting atmosphere! I was particularly close to Peter Sewell, with whom I discussed the ideas here extensively and who was generous with his guidance.

There was a trend in the community at the time of building complex process calculi (for encryption, Ambients, etc.) where the free syntax would be quotiented by a structural congruence to "stir the soup" and allow different parts of a tree to float together; reaction rules (unlabelled transitions) then would permit those agglomerated bits to react, to transform into something new.

Robin wanted to come up with a generalised framework, which he called Action Calculi, for modelling this style of process calculi. His framework would describe graph-like "soups" of atoms linked together by arcs representing binding and sharing; moreover the atoms could contain subgraphs inside of them for freezing activity (as in prefixing in the π-calculus), with the possibility of boundary crossing arcs (similarly to how ν-bound names in π-calculus can be used in deeply nested subterms).

Robin had an amazing talent for drawing beautiful graphs! He would "move" the nodes around on the chalkboard and reveal how a subgraph was in fact a reactum (the left-hand side of an unlabelled transition). In the initial phases of my Ph.D. I just tried to understand these graphs: they were so natural to draw on the blackboard! And yet, they were also so uncomfortable to use when written out in linear tree- and list-like syntax, with so many distinct concrete representations for the same graph.

Putting aside the beauty of these graphs, what was the benefit of this framework? If one could manage to embed a process calculus in Action Calculi, using the graph structure and fancy binding and nesting to represent the quotiented syntax, what then? We dreamt about a proposition along the following lines: if you represent your syntax (quotiented by your structural congruence) in Action Calculi graphs, and you represent your reaction rules as Action Calculi graph rewrites, then we will give you a congruential bisimulation for free!

Compared to CCS for example, many of the rich new process calculi lacked labelled transitions systems. In CCS, there was a clean, simple notion of labelled transitions and, moreover, bisimulation over those labelled transitions yielded a congruence: for all processes P and Q, and all process contexts $C[-]$, if $P \sim Q$, then $C[P] \sim C[Q]$. This is a key quality for a bisimulation to possess, since it allows modular reasoning about pieces of a process, something that's so much harder in a concurrent world than in a sequential one.

Returning to Action Calculi, we set out to make good on the dream that everyone gets a congruential bisimulation for free! Our idea was to find a general method to derive labelled transitions systems from the unlabelled transitions and then to prove that bisimulation built from those labelled transitions would be a congruence.

The idea was often discussed at that time that there was a duality whereby a process undergoing a labelled transition could be thought of as the environment providing a complementary context inducing the process to react. In the early labelled transition system in π-calculus for example, I recall hearing that P undergoing the input labelled transition xy could be thought of as the environment outputting payload y on channel x to enable a τ transition with P.

So I tried to formalise this notion that labelled transitions are environmental contexts enabling reaction, i.e. defining $P \overset{C[-]}{\rightarrow} P'$ to mean $C[P] \rightarrow P'$ provided that $C[-]$ was somehow "minimal", i.e., contained nothing superfluous beyond what was necessary to trigger the reaction. We wanted to get a rigorous definition of that intuitive idea. There was a long and difficult period (about 12 months) wandering through the weeds trying to define minimal contexts for Action Calculi graphs (in terms of minimal nodes and minimal arcs), but it was hugely complex, frustrating, and ugly and we seemed no closer to the original goal of achieving congruential bisimulation with these labelled transitions systems.

Eventually I stepped back from Action Calculi and started to work on a more theoretical definition of "minimal context" and we took inspiration from category theory. Robin had always viewed Action Calculi graphs as categorical arrows between objects (where the objects represented interfaces for plugging together arcs). At the time, there was much discussion of category theory in the air (for game theory); I certainly didn't understand most of it but found it interesting and inspiring.

If we imagine that processes and process-contexts are just categorical arrows (where the objects are arities) then context composition is arrow composition. Now, assuming we have a reaction rule $R \to R'$, we can define labelled transitions $P \overset{C[-]}{\to} P'$ as follows: there exists a context D such that $C[P] = D[R]$ and $P' = D[R']$. The first equality is a commuting diagram and Robin and I thought that we could formalise minimality by something like a categorical pushout! But that wasn't quite right as C and D are not the minimum pair (compared to all other candidates), but a minimal pair: there may be many incomparable minimal pairs all of which are witnesses of legitimate labelled transitions. There was again a long period of frustration eventually resolved when I reinvented "relative pushouts" (in place of pushouts). They are a simple notion in slice categories but I didn't know that until later. . . .

Having found a reasonable definition of "minimal", I worked excitedly on bisimulation, trying to get a proof of congruence: $P \sim Q$ implies $E[P] \sim E[Q]$. For weeks, I was considering the labelled transitions of $E[P] \overset{F[-]}{\to}$ and all the ways that could arise. The most interesting case is when a part of P, a part of E, and F all "conspire" together to generate a reaction. From that I was able to derive a labelled transition of P by manipulating relative pushouts, which by hypothesis yielded a labelled transition of Q, and then, via a sort of "pushout pasting", a labelled transition $E[Q] \overset{F[-]}{\to}$. It was a wonderful moment of elation when I pasted all the diagrams together on Robin's board and we realised that we had the congruence property for our synthesised labels!

We looked back again at Action Calculi, using the notion of relative pushouts to guide us (instead of the arbitrary approach we had considered before) and we further looked at other kinds of process calculi syntax to see how relative pushouts could work there. . . . Returning to the original motivation to make Action Calculi a universal framework with congruential bisimulation for free, I'm not convinced of its utility. But it was the challenge that led us to the journey of the relative pushout work, which I think is beautiful.

Davide: What influence did this work have in the rest of your career? How much of your subsequent work built on it?

James: It was thanks to this work that I visited INRIA Rocquencourt to discuss process calculi with Jean-Jacques Lévy and Georges Gonthier. They kindly invited me to spend a year as postdoc in 2001 after I finished my thesis with Robin, and I ended up staying in INRIA ever since. I didn't work on bisimulation again as a research topic, but stayed interested in concurrency and distribution for a long time, working with Peter Sewell et al. on distributed language design with module migration and rebinding, and with Cédric Fournet et al. on compiler design for automatically synthesising cryptographic protocols for high level sessions specifications.

Davide: Could you tell us about your interactions with Robin Milner? What was it like to work with him? What lessons did you learn from him?

James: I was tremendously inspired by Robin.

He would stand at his huge blackboard, his large hands covered in chalk, his bicycle clips glinting on his trousers, and he would stalk up and down the blackboard—thinking and moving. There was something theatrical and artistic about it: his thinking was done in physical movement and his drawings were dynamic as the representations of his ideas evolved across the board.

I loved his drawings. They would start simple, a circle for a node, a box for a subgraph, etc. and then develop more and more detail corresponding to his intuition. (It reminded me of descriptions I had read of Richard Feynman drawing quantum interactions.)

Sometimes I recall being frustrated because I couldn't read into his formulas everything that he wanted to convey (and we would then switch back to drawings) or I would be worried that there was an inconsistency creeping in or I just couldn't keep up, so the board sessions could be a roller coaster ride at times!

Robin worked tremendously hard and consistently. He would write out and rewrite out his ideas, regularly circulating hand written documents. He would refine over and over his diagrams. Behind his achievements there was an impressive consistency of effort.

He had a lot of confidence to carry on when the sledding was hard. He had such a strong intuition of what ought to be possible, that he was able to sustain years of effort to get there.

He was generous with praise, with credit, with acknowledgement of others' ideas. He was generous in sharing his own ideas and seemed delighted when others would pick them up and carry them forward. I've always admired his openness and lack of jealousy in sharing ideas.

In his personal life, he seemed to have real compatibility with Lucy (his wife), who also kept him grounded. I still laugh when I remember once working with him at his dining room table and Lucy announcing, "Robin, enough of the mathematics. It's time to mow the lawn!"

I visited Oxford for Lucy's funeral and recall Robin putting a brave face on his future plans; I returned a few weeks later when Robin passed away himself. I miss him greatly.

Davide: What research topics are you most interested in right now? How do you see your work develop in the future?

James: I've been interested in a totally different area, namely healthcare, for many years. I'm fascinated by how patients, and information about them, flows through the complex human and machine interactions in hospital. When looking at how these flows work, and how they don't, it's possible to see where errors arise, where blockages happen, where there are informational and visual deficits that make the job of doctors and nurses difficult. I like to think visually in terms of graphs (incrementally adding detail) and physically moving through the space where the action happens—all inspired by Robin!

Interview with Luca de Alfaro, Marco Faella, Thomas A. Henzinger, Rupak Majumdar and Mariëlle Stoelinga

In what follows, "Luca A." refers to Luca Aceto, whereas "Luca" is Luca de Alfaro.

Luca A. and Mickael: You receive the CONCUR Test-of-Time Award 2022 for your paper "The Element of Surprise in Timed Games," which appeared at CONCUR 2003[7]. In that article, you studied concurrent, two-player timed games. A key contribution of your paper is the definition of an elegant timed game model, allowing both the representation of moves that can take the opponent by surprise, as they are played "faster," and the definition of natural concepts of winning conditions for the two players—ensuring that players can win only by playing according to a physically meaningful strategy. In our opinion, this is a great example of how novel concepts and definitions can advance a research field. Could you tell us more about the origin of your model?

All authors: Mariëlle and Marco were postdocs with Luca at University of California, Santa Cruz, in that period, Rupak was a student of Tom's, and we were all in close touch, meeting very often to work together. We all had worked much on games, and an extension to timed games was natural for us to consider.

[7] See `https://pub.ist.ac.at/~tah/Publications/the_element_of_surprise_in_timed_games.pdf`).

In untimed games, players propose a move, and the moves jointly determine the next game state. In these games there is no notion of real-time. We wanted to study games in which players could decide not only the moves, but also the instant in time when to play them.

In timed automata, there is only one "player" (the automaton), which can take either a transition, or a time step. The natural generalization would be a game in which players could propose either a move, or a time step.

Yet, we were unsatisfied with this model. It seemed to us that it was different to say "Let me wait 14 seconds and reconvene. Then, let me play my King of Spades" or "Let me play my King of Spades in 14 seconds." In the first, by stopping after 14 seconds, the player is providing a warning that the card might be played. In the second, there is no such warning. In other words, if players propose either a move or a time-step, they cannot take the adversary by surprise with a move at an unanticipated instant. We wanted a model that could capture this element of surprise.

To capture the element of surprise, we came up with a model in which players propose both a move and the delay with which it is played. After this natural insight, the difficulty was to find the appropriate winning condition, so that a player could not win by stopping time.

Tom: Besides the infinite state space (region construction etc.), a second issue that is specific to timed systems is the divergence of time. Technically, divergence is a built-in Büchi condition ("there are infinitely many clock ticks"), so all safety and reachability questions about timed systems are really co-Büchi and Büchi questions, respectively. This observation had been part of my work on timed systems since the early 1990s, but it has particularly subtle consequences for timed games, where no player (and no collaboration of players) should have the power to prevent time from diverging. This had to be kept in mind during the exploration of the modeling space.

All authors: We came up with many possible winning conditions, and for each we identified some undesirable property, except for the one that we published. This is in fact an aspect that did not receive enough attention in the paper; we presented the chosen winning condition, but we did not discuss in full detail why several other conditions that might have seemed plausible did not work.

In the process of analyzing the winning conditions, we came up with many interesting games, which form the basis of many results, such as the result on lack of determinization, on the need for memory in reachability games (even when clock values are part of the state), and most famously as it gave the title to the paper, on the power of surprise.

After this fun ride came the hard work, where we had to figure out how to solve these games. We had worked at symbolic approaches to games before, and we followed the approach here, but there were many complex technical adaptations required. When we look at the paper in the distance of time, it has this combination of a natural game model, but also of a fairly sophisticated solution algorithm.

Luca A. and Mickael: Did any of your subsequent research build explicitly on the results and the techniques you developed in your award-winning paper? If so, which of your subsequent results on (timed) games do you like best? Is there any result obtained by other researchers that builds on your work and that you like in particular or found surprising?

Luca: Marco and I built Ticc, which was meant to be a tool for timed interface theories, based largely on the insights in this paper. The idea was to be able to check the compatibility of real-time systems, and automatically infer the requirements that enable two system components to work well together—to be compatible in time. We thought this would be useful for hardware or embedded systems, and especially for control systems, and in fact the application is important: there is now much successful work on the compositionality of StateFlow/Simulink models.

We used MTBDDs as the symbolic engine, and Marco and I invented a language for describing the components and we wrote by pair-programming some absolutely beautiful Ocaml code that compiled real-time component models into MTBDDs (perhaps the nicest code I have ever written). The problem was that we were too optimistic in our approach to state explosion, and we were never able to study any system of realistic size.

After this, I became interested in games more in an economic setting, and from there I veered into incentive systems, and from there to reputation systems and to a three-year period in which I applied reputation systems in practice in industry, thus losing somewhat touch with formal methods work.

Marco: I've kept working on games since the award-winning paper, in one way or another. The closest I've come to the timed game setting has been with controller synthesis games for hybrid automata. In a series of papers, we had fun designing and implementing symbolic algorithms that manipulate polyhedra to compute the winning region of a linear hybrid game. The experience gained on timed games helped me recognize the many subtleties arising in games played in real time on a continuous state-space.

Mariëlle: I have been working on games for test case generation: One player represents the tester, which chooses inputs to test; the other player represents the System-under-Test, and chooses the outputs of the system. Strategy synthesis algorithms can then compute strategies for the tester that maximize all kinds of objectives, e.g., reaching certain states, test coverage etc.

A result that I really like is that we were able to show a very close correspondence between the existing testing frameworks and game theoretic frameworks: Specifications act as game arenas; test cases are exactly game strategies, and the conformance relation used in testing (namely ioco) coincides with game refinement (i.e., alternating refinement).

Rupak: In an interesting way, the first paper on games I read was the one by Maler, Pnueli and Sifakis (STACS 1995)[8] that had both fixpoint algorithms and timed games (without "surprise"). So the problem of symbolic solutions to games and their applications in synthesis followed me throughout my career. I moved to finding controllers for games with more general (non-linear) dynamics, where we worked on abstraction techniques. We also realized some new ways to look at restricted classes of adversaries. I was always fortunate to have very good collaborators who kept my interest alive with new insights. Very recently, I have gotten interested in games from a more economic perspective, where players can try to signal each other or persuade each other about private information but it's too early to tell where this will lead.

Luca A. and Mickael: What are the research topics that you find most interesting right now? Is there any specific problem in your current field of interest that you'd like to see solved?

Mariëlle: Throughout my academic life, I have been working on stochastic analysis, with Luca and Marco, we worked on stochastic games a lot. First only on theory, but later also on industrial applications, especially in the railroad and high-tech domain. At some point in time, I realized that my work was actually centred around analysing failure probabilities and risk. That is how I moved into risk analysis; the official title of the chair I hold is Risk Management for High Tech Systems.

The nice thing is: this sells much better than Formal Methods! Almost nobody knows what Formal Methods are, and if they know, people think "yes, those difficult people who urge us to specify everything mathematically." For risk management, this is completely different: everybody understands that this is an important area.

Luca: I am currently working on computational ecology, on machine learning (ML) for networks, and on fairness in data and ML. In computational ecology, we are working on the role of habitat and territory for species viability. We use ML techniques to write "differentiable algorithms," where we can compute the effect of each input, such as the kind of vegetation in each square-kilometer of territory, on the output. If all goes well, this will enable us to efficiently compute which regions should be prioritized for protection and habitat conservation.

[8] See `https://www-verimag.imag.fr/~sifakis/RECH/Synth-MalerPnueli.pdf`.

In networks, we have been able to show that reinforcement learning can yield tremendous throughput gains in wireless protocols, and we are now starting to work on routing and congestion control.

And in fairness and ML, we have worked on the automatic detection of anomalous data subgroups (something that can be useful in model diagnostics), and we are now working on the spontaneous inception of discriminatory behavior in agent systems.

While these do not really constitute a coherent research effort, I can certainly say that I am having a grand tour of computer science—the kind of joy ride one can afford with tenure!

Rupak: I have veered between practical and theoretical problems. I am working on charting the decidability frontier for infinite-state model checking problems (most recently, for asynchronous programs and context-bounded reachability). I am also working on applying formal methods to the world of cyber-physical systems—mostly games and synthesis. Finally, I have become very interested in applying formal methods to large scale industrial systems through a collaboration with Amazon Web Services. There is still a large gap between what is theoretically understood and what is practically applicable to these systems; and the problems are a mix of technical and social.

Luca A. and Mickael: You have a very strong track record in developing theoretical results and in applying them to real-life problems. In our, admittedly biased, opinion, your work exemplifies Ben Schneiderman's Twin-Win Model, which propounds the pursuit of "the dual goals of breakthrough theories in published papers and validated solutions that are ready for widespread dissemination." Could you say a few words on your research philosophy? How do you see the interplay between basic and applied research?

Luca: This is very kind for you to say, and a bit funny to hear, because certainly when I was young I had a particular talent for getting lost in useless theoretical problems.

I think two things played in my favor. One is that I am curious. The other is that I have a practical streak: I still love writing code and tinkering with "things," from IoT to biology to web and more. This tinkering was at the basis of many of the works I did. My work on reputation systems started when I created a wiki on cooking; people were vandalizing it, and I started to think about game theory and incentives for collaboration, which led to my writing much of the code for Wikipedia analysis, and at Google, for Maps edits analysis. My work on networks started with me tinkering with simple reinforcement-learning schemes that might work, and writing the actual code. On the flip side, my curiosity too often had the better of me, so that I have been unable to pay the continuous and devoted attention to a single research field. I am not a specialist in any single thing I do or

I have done. I am always learning the ropes of something I don't quite know yet how to do.

My applied streak probably gave me some insight on which problems might be of more practical relevance, and my frequent field changes have allowed me to bring new perspectives to old problems. There were not many people using reinforcement learning for wireless networks, there are not many who write ML and GPU code and also avidly read about conservation biology.

Rupak: I must say that Tom and Luca were very strong influencers for me in my research: both in problem selection and in appreciating the joy of research. I remember one comment of Tom, paraphrased as "Life is short. We should write papers that get read." I spent countless hours in Luca's office and learnt a lot of things about research, coffee, the ideal way to make pasta, and so on.

Marco: It was an absolute privilege to be part of the group that wrote that paper (my 4th overall, according to DBLP). I'd like to thank my coauthors, and Luca in particular, for guiding me during those crucially formative years.

Mariëlle: I fully agree!

Luca A. and Mickael: Several of you have high-profile leadership roles at your institutions. What advice would you give to a colleague who is about to take up the role of department chair, director of a research centre, dean or president of a university? How can one build a strong research culture, stay research active and live to tell the tale?

Luca: My colleagues may have better advice; my productivity certainly decreased when I was department chair, and is lower even now that I am the vice-chair. When I was young, I was ambitious enough to think that my scientific work would have the largest impact among the things I was doing. But I soon realized that some of the greatest impact was on others: on my collaborators, on the students I advised, who went on to build great careers and stayed friends, and on all the students I was teaching. This awareness serves to motivate and guide me in my administrative work. The Computer Science department at University of California, Santa Cruz, is one of the ten largest in the number of students we graduate, and the time I spend on improving its organization and the quality of the education it delivers is surely very impactful. My advice to colleagues is to consider their service not as an impediment to research, but as one of the most impactful things they do.

My way of staying alive is to fence off some days that I only dedicate to research (aside from some unavoidable emergency), and also, to have collaborators that give me such joy in working together that they brighten and energize my whole day.

Luca A. and Mickael: Finally, what advice would you give to a young researcher who is keen to start working on topics related to concurrency theory today?

Luca: Oh that sounds very interesting! And, may I show you this very interesting thing we are doing in Jax to model bird dispersal? We feed in this climate and vegetation data, and then we. . . .

Just kidding. Just kidding. If I come to CONCUR I promise not to lead any of the concurrency yearlings astray. At least I will try.

My main advice would be this: work on principles that allow correct-by-design development. If you look at programming languages and software engineering, the progress in software productivity has not happened because people have become better at writing and debugging code written in machine language or C. It has happened because of the development of languages and software principles that make it easier to build large systems that are correct by construction. We need the same kind of principles, (modeling) languages, and ideas to build correct concurrent systems. Verification alone is not enough. Work on design tools, ideas to guide design, and design languages.

Tom: In concurrency theory we define formalisms and study their properties. Most papers do the studying, not the defining: they take a formalism that was defined previously, by themselves or by someone else, and study a property of that formalism, usually to answer a question that is inspired by some practical motivation. To me, this omits the most fun part of the exercise, the *defining* part. The point I am trying to make is not that we need more formalisms, but that, if one wishes to study a specific question, it is best to study the question on the simplest possible formalism that exhibits exactly the features that make the question meaningful. To do this, one often has to define that formalism. In other words, the formalism should follow the question, not the other way around. This principle has served me well again and again and led to formalisms such as timed games, which try to capture the essence needed to study the power of timing in strategic games played on graphs. So my advice to a young researcher in concurrency theory is: choose your formalism wisely and don't be afraid to define it.

Rupak: Problems have different measures. Some are practically justified ("Is this practically relevant in the near future?") and some are justified by the foundations they build ("Does this avenue provide new insights and tools?"). Different communities place different values on the two. But both kinds of work are important and one should recognize that one set of values is not universally better than the other.

Mariëlle: As Michael Jordan puts it: "Just play. Have fun. Enjoy the game."

European
Association for
Theoretical
Computer
Science

E A T C S

EATCS

HISTORY AND ORGANIZATION

EATCS is an international organization founded in 1972. Its aim is to facilitate the exchange of ideas and results among theoretical computer scientists as well as to stimulate cooperation between the theoretical and the practical community in computer science.

Its activities are coordinated by the Council of EATCS, which elects a President, Vice Presidents, and a Treasurer. Policy guidelines are determined by the Council and the General Assembly of EATCS. This assembly is scheduled to take place during the annual **I**nternational **C**olloquium on **A**utomata, **L**anguages and **P**rogramming (ICALP), the conference of EATCS.

MAJOR ACTIVITIES OF EATCS

- Organization of ICALP;
- Publication of the "Bulletin of the EATCS;"
- Award of research and academic career prizes, including the EATCS Award, the Gödel Prize (with SIGACT), the Presburger Award, the EATCS Distinguished Dissertation Award, the Nerode Prize (joint with IPEC) and best papers awards at several top conferences;
- Active involvement in publications generally within theoretical computer science.

Other activities of EATCS include the sponsorship or the cooperation in the organization of various more specialized meetings in theoretical computer science. Among such meetings are: CIAC (Conference of Algorithms and Complexity), CiE (Conference of Computer Science Models of Computation in Context), DISC (International Symposium on Distributed Computing), DLT (International Conference on Developments in Language Theory), ESA (European Symposium on Algorithms), ETAPS (The European Joint Conferences on Theory and Practice of Software), LICS (Logic in Computer Science), MFCS (Mathematical Foundations of Computer Science), WADS (Algorithms and Data Structures Symposium), WoLLIC (Workshop on Logic, Language, Information and Computation), WORDS (International Conference on Words).

Benefits offered by EATCS include:

- Subscription to the "Bulletin of the EATCS;"
- Access to the Springer Reading Room;
- Reduced registration fees at various conferences;
- Reciprocity agreements with other organizations;
- 25% discount when purchasing ICALP proceedings;
- 25% discount in purchasing books from "EATCS Monographs" and "EATCS Texts;"
- Discount (about 70%) per individual annual subscription to "Theoretical Computer Science;"
- Discount (about 70%) per individual annual subscription to "Fundamenta Informaticae."

Benefits offered by EATCS to Young Researchers also include:

- Database for Phd/MSc thesis
- Job search/announcements at Young Researchers area

(1) THE ICALP CONFERENCE

ICALP is an international conference covering all aspects of theoretical computer science and now customarily taking place during the second or third week of July. Typical topics discussed during recent ICALP conferences are: computability, automata theory, formal language theory, analysis of algorithms, computational complexity, mathematical aspects of programming language definition, logic and semantics of programming languages, foundations of logic programming, theorem proving, software specification, computational geometry, data types and data structures, theory of data bases and knowledge based systems, data security, cryptography, VLSI structures, parallel and distributed computing, models of concurrency and robotics.

Sites of ICALP meetings:

- Paris, France 1972
- Saarbrücken, Germany 1974
- Edinburgh, UK 1976
- Turku, Finland 1977
- Udine, Italy 1978
- Graz, Austria 1979
- Noordwijkerhout, The Netherlands 1980
- Haifa, Israel 1981
- Aarhus, Denmark 1982
- Barcelona, Spain 1983
- Antwerp, Belgium 1984
- Nafplion, Greece 1985
- Rennes, France 1986
- Karlsruhe, Germany 1987
- Tampere, Finland 1988
- Stresa, Italy 1989
- Warwick, UK 1990
- Madrid, Spain 1991
- Wien, Austria 1992
- Lund, Sweden 1993
- Jerusalem, Israel 1994
- Szeged, Hungary 1995
- Paderborn, Germany 1996
- Bologne, Italy 1997
- Aalborg, Denmark 1998
- Prague, Czech Republic 1999
- Genève, Switzerland 2000
- Heraklion, Greece 2001
- Malaga, Spain 2002
- Eindhoven, The Netherlands 2003
- Turku, Finland 2004
- Lisabon, Portugal 2005
- Venezia, Italy 2006
- Wrocław, Poland 2007
- Reykjavik, Iceland 2008
- Rhodes, Greece 2009
- Bordeaux, France 2010
- Zürich, Switzerland 2011
- Warwick, UK 2012
- Riga, Latvia 2013
- Copenhagen, Denmark 2014
- Kyoto, Japan 2015
- Rome, Italy 2016
- Warsaw, Poland 2017
- Prague, Czech Republic 2018
- Patras, Greece 2019
- Saarbrücken, Germany (virtual conference) 2020
- Glasgow, UK (virtual conference) 2021

(2) THE BULLETIN OF THE EATCS

Three issues of the Bulletin are published annually, in February, June and October respectively. The Bulletin is a medium for *rapid* publication and wide distribution of material such as:

- EATCS matters;
- Technical contributions;
- Columns;
- Surveys and tutorials;
- Reports on conferences;
- Information about the current ICALP;
- Reports on computer science departments and institutes;
- Open problems and solutions;
- Abstracts of Ph.D. theses;
- Entertainments and pictures related to computer science.

Contributions to any of the above areas are solicited, in electronic form only according to formats, deadlines and submissions procedures illustrated at `http://www.eatcs.org/bulletin`. Questions and proposals can be addressed to the Editor by email at `bulletin@eatcs.org`.

(3) OTHER PUBLICATIONS

EATCS has played a major role in establishing what today are some of the most prestigious publication within theoretical computer science.

These include the *EATCS Texts* and the *EATCS Monographs* published by Springer-Verlag and launched during ICALP in 1984. The Springer series include *monographs* covering all areas of theoretical computer science, and aimed at the research community and graduate students, as well as *texts* intended mostly for the graduate level, where an undergraduate background in computer science is typically assumed.

Updated information about the series can be obtained from the publisher.

The editors of the EATCS Monographs and Texts are now M. Henzinger (Vienna), J. Hromkovič (Zürich), M. Nielsen (Aarhus), G. Rozenberg (Leiden), A. Salomaa (Turku). Potential authors should contact one of the editors.

EATCS members can purchase books from the series with 25% discount. Order should be sent to:

Prof.Dr. G. Rozenberg, LIACS, University of Leiden,
P.O. Box 9512, 2300 RA Leiden, The Netherlands

who acknowledges EATCS membership and forwards the order to Springer-Verlag.

The journal *Theoretical Computer Science*, founded in 1975 on the initiative of EATCS, is published by Elsevier Science Publishers. Its contents are mathematical and abstract in spirit, but it derives its motivation from practical and everyday computation. Its aim is to understand the nature of computation and, as a consequence of this understanding, provide more efficient methodologies.

The Editor-in-Chief of the journal currently are D. Sannella (Edinburgh), L. Kari and P.G. Spirakis (Patras).

ADDITIONAL EATCS INFORMATION

For further information please visit `http://www.eatcs.org`, or contact the President of EATCS:

Prof. Artur Czumaj,
Email: *`president@eatcs.org`*

EATCS MEMBERSHIP

DUES

The dues are €40 for a period of one year (two years for students / Young Researchers). Young Researchers, after paying, have to contact `secretary@eatcs.org`, in order to get additional years. A new membership starts upon registration of the payment. Memberships can always be prolonged for one or more years.

In order to encourage double registration, we are offering a discount for SIGACT members, who can join EATCS for €35 per year. We also offer a five-euro discount on the EATCS membership fee to those who register both to the EATCS and to one of its chapters. Additional €35 fee is required for ensuring the *air mail* delivery of the EATCS Bulletin outside Europe.

HOW TO JOIN EATCS

You are strongly encouraged to join (or prolong your membership) directly from the EATCS website `www.eatcs.org`, where you will find an online registration form and the possibility of secure online payment. Alternatively, contact the Secretary Office of EATCS:

Mrs. Efi Chita,
Computer Technology Institute & Press (CTI)
1 N. Kazantzaki Str., University of Patras campus,
26504, Rio, Greece
Email: `secretary@eatcs.org`*,*
Tel: +30 2610 960333, Fax: +30 2610 960490

If you are an EATCS member and you wish to prolong your membership or renew the subscription you have to use the Renew Subscription form. The dues can be paid via paypal and all major credit cards are accepted.

For adittional information please contact the Secretary of EATCS:

Prof. Emanuela Merelli
via Madonna delle Carceri, 9
Computer Science Build. 1st floor
University of Camerino,
Camerino 62032, Italy
Email: `secretary@eatcs.org`*,*
Tel: +39 0737402567